MIXED INTEGER BILINEAR PROGRAMMING WITH APPLICATIONS TO THE POOLING PROBLEM

A Thesis
Presented to
The Academic Faculty

by

Akshay Gupte

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in
Operations Research

H. Milton Stewart School of Industrial and Systems Engineering
Georgia Institute of Technology
December 2012

MIXED INTEGER BILINEAR PROGRAMMING WITH APPLICATIONS TO THE POOLING PROBLEM

Approved by:

Professor Shabbir Ahmed, Advisor
H. Milton Stewart School of Industrial and Systems Engineering
Georgia Institute of Technology

Asst. Professor Santanu Dey, Advisor
H. Milton Stewart School of Industrial and Systems Engineering
Georgia Institute of Technology

Professor George Nemhauser
H. Milton Stewart School of Industrial and Systems Engineering
Georgia Institute of Technology

Associate Professor Joel Sokol
H. Milton Stewart School of Industrial and Systems Engineering
Georgia Institute of Technology

Dr. Myun Seok Cheon
Corporate Strategic Research
ExxonMobil Research and Engineering Company

Date Approved: August 1, 2012

To my beloved wife Akshi.

ACKNOWLEDGEMENTS

I take this opportunity to thank my advisors, Prof. Shabbir Ahmed and Asst. Prof. Santanu Dey, for their guidance and support throughout my Ph.D. studies. They have provided invaluable advice while conducting research and inspired me to pursue an academic career. Thanks to the rest of my committee, Prof. George Nemhauser, Assoc. Prof. Joel Sokol, and Dr. Myun Seok Cheon for their helpful comments and suggestions.

I also appreciate the role of the faculty and staff at the School of Industrial and Systems Engineering in providing me with a high quality education. The seemingly infinite array of courses kept me busy along with my research. Dr. Gary Parker and Ms. Pam Morrison were a great help with administrative issues.

Special thanks to my wife, Akshi, for her unconditional love and relentless support during these many years of graduate school. No doubt the journey has been long and arduous, and at times very stressful, however her comforting presence made matters easier for me. I deeply appreciate all the sacrifices she has made to help me achieve my goals. I also thank my parents for presenting me with the opportunity to attend graduate school and for their constant encouragement.

I had like to thank my peers at Georgia Tech - Dimitri, Feng, Ahmed, Gustavo, Diego, Pete, Qie, Steve, and many more. Doing research becomes ϵ-easier when you have a good support group to exchange ideas and blow off steam outside work. Special mention also goes to the intangibles, such as the Pandoras and Spotifys, whose constant internet streaming enabled writing parts of this document into the wee hours.

Finally, I would like to acknowledge the financial support for this research from Exxon-Mobil Corporate Strategic Research. I also thank Dr. Ahmet Keha from ExxonMobil for his guidance during my summer internship. Thanks to School of Industrial and Systems Engineering for John Morris fellowship and graduate assistantships during the initial few semesters.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

This dissertation studies mixed integer bilinear programming (MIBLP) problems, which form a class of optimization problems defined as

$$\begin{aligned} \min_{x,y} \quad & x^\top Q_0 y + f_0^\top x + g_0^\top y \\ \text{s.t.} \quad & Ax + Gy \leq h_0 \\ & x^\top Q_t y + f_t^\top x + g_t^\top y \leq h_t, \qquad t = 1, \ldots, p, \\ & x, y \geq \mathbf{0},\ x_i \in \mathbb{Z}_+, i \in \mathcal{I},\ y_j \in \mathbb{Z}_+, j \in \mathcal{J}. \end{aligned}$$

This problem has two sources of nonconvexity: one due to integer variables and second due to bilinear functions of the form $x^\top Q y + f^\top x + g^\top y$. Hence, a MIBLP must be solved using a global optimization algorithm to obtain a exact solution. A central theme of this research is the use of mixed integer linear programming (MILP)-type techniques for solving MIBLPs.

Mixed integer bilinear programs find many applications, a particular one being the pooling problem. Chapter 1 introduces the pooling problem as a minimum cost network flow problem on a directed graph. The classical pooling problem is a continuous bilinear program (BLP). We review various equivalent optimization models for this problem and also address conventional relaxations obtained by relaxing all the bilinear terms. We compare the strengths of these relaxations and provide stronger than previously-known results about the tightness of the various problem formulations.

The main contributions of this dissertation follow in Chapters 2 – 4. In Chapter 2, we investigate relaxations of a single node flow substructure of general network flow problems that involve material mixing phenomenon at certain nodes in the graph. We study some basic properties and generate disjunctive representations from appropriate restrictions of this set. Valid linear inequalities are derived by extending the theory of lifting restrictions of a set from MILP literature. We present explicit expressions for a exponential class of valid inequalities. To the best of our knowledge, this is the first study that contributes a

polyhedral relaxation for this set without adding any auxiliary variables.

Chapter 3 studies general MIBLPs where every bilinear term is expressed as the product of one continuous and one integer variable. We propose using a mixed $\{0, 1\}$ MILP formulation for solving this problem. Facet-defining inequalities are presented for the convex hull of the MILP reformulation a single mixed integer bilinear term. The proposed cutting planes are tested on five classes of instances. Our experiments suggest that these cutting planes can be very effective in solving MIBLPs where all bilinear terms appear in the objective function. We also provide extensions of our approach to bounded bilinear terms and more generalized representations of a integer variable.

Finally in Chapter 4, we study different ways of discretizing the pooling problem and solving the discretized problem as a MILP. The discretized problem is a MIBLP and a restriction of the original problem that provides feasible solutions and upper bounds on the global optimum of the pooling problem. We address different variable discretization schemes. Next, we propose a new MILP discretization that has a network flow interpretation. The emphasis of this chapter is on empirically testing the performance of the different MILP models. Our experiments show that on a certain class of pooling problems, discretization outperforms global optimization solvers in finding good quality feasible solutions to the problem. On many of the test instances, our proposed network flow MILP provides the best solution.

CHAPTER I

POOLING PROBLEM

1.1 Introduction

The classical minimum cost network flow problem seeks to find the optimal way of sending raw materials from a set of suppliers to a set of customers via certain transshipment nodes in a directed capacitated network. The blending problem, which typically arises in refinery processes in the petroleum industry, is a type of minimum cost network flow problem with only two sets of nodes: suppliers and customers. The raw material at each supplier possesses multiple specifications, examples being concentrations of chemical compounds such as sulphur, carbon, or physical properties such as density, octane number. End products for the customers are created by directly mixing raw materials available from different suppliers. The mixing process should occur in a way such that the end products contain a certain minimum and/or maximum level of each specification. As in the network flow problem, the objective in the blending problem is to minimize the total cost of producing demand.

The pooling problem, a generalization of the blending problem, combines features of both the classical network flow problem and the blending problem and can be stated in informal terms as follows: Given a list of available suppliers (inputs) with raw materials containing known specifications, find the minimum cost way of mixing these materials in intermediate tanks (pools) so as to meet the demand and specification requirements at multiple final blends (outputs). Thus in a pooling problem, flows are blended in two stages: first the raw materials are allowed to be mixed in intermediate tanks referred to as pools and then sent forth from the pools to be mixed again at the output to form end products. The need for mixing raw materials at pools occurs if the output requirements for demand and specification level cannot be satisfied directly by mixing the inputs or due to other logical considerations such as economies of scale, etc. It is also possible to send some of the flow directly from inputs to the outputs. Figure 1 illustrates the pooling problem as a network

flow problem over three sets of nodes: inputs, pools (or transshipment), and outputs.

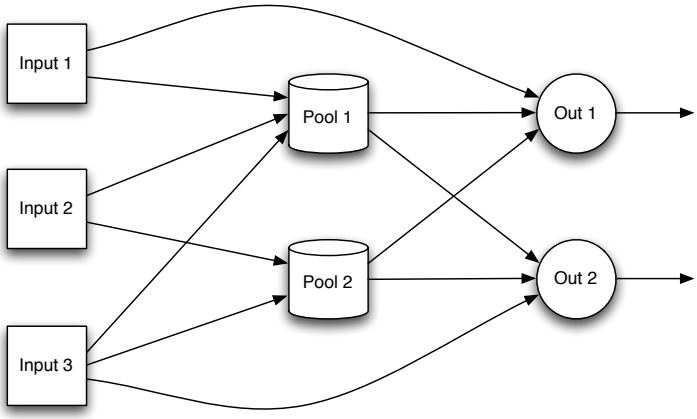


Figure 1: A sample pooling problem

The inflows, outflows, and specification values at each pool are decision variables in the optimization model. Constraints that track specification level at each pool and that determine the level of specification available at each output are formulated as bilinear constraints. As a result, the pooling problem is a bilinear program (BLP), which is a particular case of a nonconvex quadratic program with quadratic constraints (QCQP), and hence must be solved using a global optimization algorithm to obtain a exact solution. In contrast, the classical blending problem, due to the absence of pools, can be formulated as a linear program (LP).

The pooling problem was first proposed by Haverly [60]. Early efforts in solving this problem were based on recursive LP [60] and successive LP [15] methods. An algorithm based on generalized Benders' decomposition was proposed by Floudas and Aggarwal [40]. Sensitivity analysis was carried out by Greenberg [51] and Frimannslund et al. [45]. All these methods could not address the issue of convergence to a global optimal solution. An empirical comparison of various local optimization solvers was performed by Poku et al. [84]. More recently, many global optimization algorithms have been proposed. Visweswaran and Floudas [111] used duality theory and Lagrangian relaxations to develop the GOP algorithm, which is applicable to a wide range of nonconvex nonlinear programs (NLPs). Ben-Tal et al. [25] proposed another duality-related approach by defining an alternative

formulation for the pooling problem. More studies in Lagrangian-based methods are found in Adhya et al. [3] and Almutairi and Elhedhli [9]. Foulds et al. [43] applied the branch-and-bound algorithm of Al-Khayyal and Falk [5], designed for bilinear programs, to solve pooling problems. Later, Quesada and Grossmann [85] extended this approach to general chemical process network problems with bilinear terms. Audet et al. [14] solved pooling problems using a branch-and-cut algorithm developed for nonconvex QCQPs. A general branch-and-bound algorithm to solve nonconvex NLPs to global optimality was proposed by Horst and Tuy [61], whose variants are widely incorporated in global optimizers such as BARON by Sahinidis [93] and Couenne by Belotti et al. [23]. A dedicated solver for pooling problems was recently implemented by Misener et al. [78].

In the remainder of this chapter, we review various optimization models for the pooling problem and formally prove their equivalence. We also present some variants of the pooling problem that involve additional constraints. The second half of the chapter addresses conventional relaxations for the problem. These relaxations are obtained by introducing convex and concave envelopes of each bilinear term that arises in the problem formulation. We analytically compare the strengths of these relaxations. Our results generalize previous results on relaxations of the pooling problem.

We close this introduction by commenting that although the study of pooling problems was motivated using the example of refinery processes, the problem also finds applications in other areas such as wastewater treatment [64], emissions regulation [46], agricultural industry [26], etc. More industrial applications can be found in Amos et al. [10], DeWitt et al. [37], Kallrath [63], Visweswaran [110]. The reader is referred to Audet et al. [14], Haugland [59], Misener and Floudas [74], Tawarmalani and Sahinidis [101] for previous surveys on the pooling problem.

1.2 Problem Formulations

This section formally defines the pooling problem as a type of a bilinear network flow problem on an arbitrary directed graph. First let us define several parameters that will be useful in stating the pooling problem in mathematical terms.

1.2.1 Model parameters

Consider a directed graph $G = (\mathcal{N}, \mathcal{A})$ where $\mathcal{N}$ is the set of nodes and $\mathcal{A}$ the set of arcs. $\mathcal{N}$ can be partitioned into three nonempty subsets $I, L, J \subset \mathcal{N}$. Here I denotes the set of inputs, L the set of pools, and J the set of outputs. We assume that $\mathcal{A} \subseteq (I \times L) \cup (L \times L) \cup (L \times J) \cup (I \times J)$, i.e. there are no arcs between two inputs or two outputs and no backward arcs from pools to inputs or outputs to inputs or outputs to pools. We also assume that every pool has both in-degree and out-degree of at least 1. Similarly, every input (output) has out-degree (in-degree) at least 1. Otherwise, the corresponding nodes can be simply eliminated from the problem. Note that we have allowed the presence of pool-pool arcs in the set $\mathcal{A}$. Traditionally, problem instances with $\mathcal{A} \cap (L \times L) = \emptyset$ are referred to as *standard pooling problems* and as *generalized pooling problems*, otherwise. Unless stated explicitly, we do not differentiate between these two cases since our aim is to present a more unified treatment for all classes of pooling problems. For every pool $l \in L$, define $I_l \subseteq I$ as the subset of inputs from which there exists a directed path to l in G. Define $L_I := \{l \in L \colon \nexists l' \in L \text{ s.t. } (l', l) \in \mathcal{A}\}$ to be the set of nodes in L with incoming arcs only from nodes in I.

We first state a simple fact about G that follows from topological sorting of directed acyclic graphs.

Observation 1.1. *If G is acyclic, then $L_I \neq \emptyset$.*

It is easy to verify that the absence of directed cycles in G is not a necessary condition for L_I to be nonempty.

Assumption 1.1. G is acyclic.

Let K denote the set of specifications that are tracked across the pooling problem. For each arc $(i, j) \in \mathcal{A}$, let c_{ij} be the variable cost of sending a unit flow on this arc. For every node $i \in \mathcal{N}$, let C_i be the capacity of this node, i.e. the maximum amount of incoming or outgoing flow from node i. For a pool $l \in L$, its capacity C_l can be interpreted as the volumetric size of the pool tank, whereas for input $i \in I$, C_i is the total available supply

and for $j \in J$, C_j is the maximum demand. The upper bound on flow on arc $(i,j) \in \mathcal{A}$ is denoted as u_{ij}. Typically, the upper bounds on flows are such that the capacities of adjacent nodes are not violated, i.e. $u_{ij} = \min\{C_i, C_j\}, (i,j) \in \mathcal{A}$. However we allow the arcs in G to carry arbitrary upper bounds. λ_{ik} denotes the level of specification k in raw material at input i, for all $i \in I$ and $k \in K$. Likewise, $\mu_{jk}^{\min}$ and $\mu_{jk}^{\max}$ are the lower and upper bound requirements on level of specification k at output j, for all $j \in J$ and $k \in K$. We assume that each of the values $\sum_k \lambda_{ik}, \sum_k \mu_{jk}^{\min}, \sum_k \mu_{jk}^{\max}$ is between $[0,1]$ for all $i \in I, j \in J$. This assumption is without loss of generality since the given values for these parameters can always be normalized.

Let y_{ij} be the flow on arc $(i,j) \in \mathcal{A}$.

Assumption 1.2. For notational simplicity, we will always write equations using the flow variables y_{ij} with the understanding that y_{ij} is defined only for $(i,j) \in \mathcal{A}$.

At each pool $l \in L$, the total amount of incoming flow must equal the total amount of outgoing flow.

$$\sum_{i \in I \cup L} y_{il} = \sum_{j \in L \cup J} y_{lj}, \quad l \in L. \tag{1}$$

The capacity constraints at each node in G are stated as

$$\sum_{j \in L \cup J} y_{ij} \leq C_i, \quad i \in I, \tag{2a}$$

$$\sum_{j \in L \cup J} y_{lj} \leq C_l, \quad l \in L, \tag{2b}$$

$$\sum_{i \in I \cup L} y_{ij} \leq C_j, \quad j \in J. \tag{2c}$$

Finally, flows on G are bounded by individual arc capacities.

$$0 \leq y_{ij} \leq u_{ij}, \quad (i,j) \in \mathcal{A}. \tag{3}$$

Denote $\mathcal{F} := \{y \in \Re_+^{|\mathcal{A}|} \colon (1) - (3)\}$ as the polyhedral set that defines feasible flows on G. Additional constraints can be included in $\mathcal{F}$, such as minimum supply (demand) at input (output), respectively.

1.2.2 Concentration model : p-formulation

In the pooling problem we send flows from inputs, mix them in pool tanks, and finally send the mixture from pools to outputs. Thus the mixtures in each pool and output carry specifications whose concentration values, denoted by p_{jk} for $j \in L \cup J, k \in K$, can be determined as

$$p_{jk} = \begin{cases} \dfrac{\sum_{i \in I} \lambda_{ik} y_{ij} + \sum_{l \in L} p_{lk} y_{lj}}{\sum_{i \in I \cup L} y_{ij}} & \text{if } \sum_{i \in I \cup L} y_{ij} > 0 \\ 0 & \text{if } \sum_{i \in I \cup L} y_{ij} = 0. \end{cases}$$

Since $0 \leq \sum_{k \in K} \lambda_{ik} \leq 1$, it follows by recursion that $0 \leq \sum_{k \in K} p_{jk} \leq 1$, for $j \in L \cup J, k \in K$. Observe that the above expression for p_{jk} can be equivalently rewritten in the following bilinear form,

$$\sum_{i \in I} \lambda_{ik} y_{il} + \sum_{l' \in L} p_{l'k} y_{l'l} = p_{lk} \sum_{j \in L \cup J} y_{lj}, \quad \forall l \in L, k \in K \tag{4a}$$

$$\sum_{i \in I} \lambda_{ik} y_{ij} + \sum_{l \in L} p_{lk} y_{lj} = p_{jk} \sum_{i \in I \cup L} y_{ij}, \quad \forall j \in J, k \in K \tag{4b}$$

The bilinear equalities in (4) will be referred to as the *spec tracking constraints* since they help determine the concentration values of specifications at each pool and output.

The classical pooling problem assumes that the mixing process follows linear blending, i.e. the total amount of specification at a node is simply the sum of product of specification concentration value and total flow on each input arc into this node. More general mixing processes that occur in specialized applications where this assumption may not hold true are discussed in the survey of Misener and Floudas [74]. We will assume linear blending at pools and outputs throughout this thesis.

We are now ready to formally state the pooling problem.

Definition 1.1 (Pooling problem)**.** Given any directed graph G and its attributes, find a minimum cost feasible flow $y \in \mathcal{F}$ such that there exist some concentration values $p \in$

$\Re^{(|L|+|J|)\times|K|}$ that satisfy (4) and $\mu_{jk}^{\min} \le p_{jk} \le \mu_{jk}^{\max}$ for all $j \in J, k \in K$.

$$\begin{aligned} \min_{y,p} \quad & \sum_{(i,j)\in\mathcal{A}} c_{ij} y_{ij} \\ \text{s.t.} \quad & y \in \mathcal{F} \\ & (4),\ \mu_{jk}^{\min} \le p_{jk} \le \mu_{jk}^{\max}, \quad j \in J, k \in K. \end{aligned} \tag{Pooling}$$

For each output $j \in J$ and spec $k \in K$, we can combine the spec tracking constraints (4) and spec level requirements $\mu_{jk}^{\min} \le p_{jk} \le \mu_{jk}^{\max}$ to give bilinear inequality constraints of the form

$$\sum_{i\in I} \lambda_{ik} y_{ij} + \sum_{l\in L} p_{lk} y_{lj} \;\le\; \mu_{jk}^{\max} \sum_{i\in I\cup L} y_{ij}, \quad j \in J, k \in K, \tag{5a}$$

$$\sum_{i\in I} \lambda_{ik} y_{ij} + \sum_{l\in L} p_{lk} y_{lj} \;\ge\; \mu_{jk}^{\min} \sum_{i\in I\cup L} y_{ij}, \quad j \in J, k \in K. \tag{5b}$$

Note that the spec tracking constraints corresponding to the pools are retained. We next discuss how to use the problem structure to enforce tight bounds on the p variables at each pool.

Bound tightening at the pools. From Definition 1.1, it follows that all the flows in G originate at some input $i \in I$. Then clearly, $p_{lk} \ge 0$ since we assumed $\lambda_{ik} \ge 0$. Define

$$\underline{p}_{lk} := \min\{\lambda_{ik} \colon i \in I_l\}, \quad \bar{p}_{lk} := \max\{\lambda_{ik} \colon i \in I_l\}, \quad l \in L, k \in K. \tag{6}$$

Due to the linear blending assumption in the tracking constraints (4a), the concentration of specification k within any flow arriving at pool l must be no more than the maximum concentration over all the contributing inputs. This implies $0 \le p_{lk} \le \bar{p}_{lk}$. We now argue that $\underline{p}_{lk} \le p_{lk}$ is a valid lower bound. Consider a pool $l \in L_I$ and suppose that $p_{lk} = 0$ but $\underline{p}_{lk} > 0$ for some $k \in K$. Since all the incoming arcs to l are from I, this can only happen if $y_{il} = 0, \forall i \in I$. Flow balance (1) implies that $y_{lj} = 0, \forall j \in L \cup J$. Hence we can safely set $p_{lk} = \underline{p}_{lk}$ without violating (4a). Also, since the objective function coefficient on p_{lk} is zero, this modified solution bears the same cost as the original solution. Now consider a subset $\mathcal{L} \subseteq L \setminus L_I$. The subgraph of G induced by $\mathcal{L}$ is a directed acyclic graph. Observation 1.1 tells us that there must be some $l \in \mathcal{L}$ such that $\nexists l' \in \mathcal{L}$ with $(l', l) \in \mathcal{A}$. Since we have

already rounded up the values of $p_{l'k}$ for $l' \in L_I$, we can use similar arguments as before to conclude that $p_{lk} \geq \underline{p}_{lk}$ is valid. Finally, an induction on the size of $\mathcal{L}$ completes our claim that $p_{lk} \geq \underline{p}_{lk}$ is valid for all $l \in L, k \in K$.

Thus we have the following optimization model ($\mathbb{P}$), commonly referred to as the *p*-formulation.

$$\begin{aligned} \min_{y,p} \quad & \sum_{(i,j)\in\mathcal{A}} c_{ij} y_{ij} \\ \text{s.t.} \quad & y \in \mathcal{F}, \ (6) \\ & (4a), \ (5) \end{aligned} \tag{$\mathbb{P}$}$$

Complexity. Recently, Alfaki and Haugland [6] provided a formal proof for the NP-hardness of the pooling problem via a reduction from the maximum stable set problem to the standard pooling problem with a single pool. They also gave a recursive algorithm that runs in polynomial time for fixed $|K|$ and solves a single pool problem with no direct arcs from inputs to outputs. Observe that the sole purpose of having variables p_{lk} in ($\mathbb{P}$) is to enforce that all the outgoing arcs from a pool carry the same concentration value for a spec. Consider a pooling problem where $|\{j \in \mathcal{N} \colon (l,j) \in \mathcal{A}\}| = 1$ for $l \in L$. Substitute a new variable w_{lkj} for bilinear terms $p_{lk}y_{lj}$ in (4a) and (5). Since each pool has only one outgoing arc, we need not enforce the spec consistency constraints $w_{lkj} = p_{lk}y_{lj}$. Thus the formulation ($\mathbb{P}$) can be completely linearized in this special case and solved as a single LP in polynomial time.

1.2.3 Alternate formulations

1.2.3.1 Proportion model : q-formulation

The *q*-formulation for the standard pooling problems was proposed by Ben-Tal et al. [25]. In this formulation, Ben-Tal et al. modeled (5) using proportion variables q_{il}, for $l \in L, i \in I$, which denote the fraction of incoming flow to pool l that is contributed by input i. Thus,

$$\begin{aligned} \sum_{i\in I} q_{il} &= 1, \\ y_{il} &= q_{il} \sum_{i'\in I} y_{i'l} = q_{il} \sum_{j\in L\cup J} y_{lj}. \end{aligned}$$

Then, in the case of standard pooling problems, the specification tracking constraints (4) imply

$$p_{lk} = \sum_{i \in I} \lambda_{ik} q_{il}, \quad l \in L, k \in K.$$

Alfaki and Haugland [7] developed a q-formulation for generalized pooling problems that has bilinear terms of the form $q_{il}y_{lj}$ with $\mathcal{O}(|I||L|)$ proportion variables. We discuss the intuition behind this formulation.

For every pool $l \in L$, define I_l as the subset of inputs from which there exists a directed path to l in G. Let q_{il} denote the fraction of incoming flow to pool $l \in L$ that originated from input $i \in I_l$. Note that in this definition of the proportion variable q_{il}, we do not distinguish between flows that started at i and reached l along different paths. By definition, the q's must sum to 1 across all inputs and hence we have

$$q_l \in \Delta^{|I_l|} := \{q_l \geq 0 \colon \sum_{i \in I_l} q_{il} = 1\}, \quad l \in L, \tag{7}$$

where q_l is the vector $(q_{il})_{i \in I_l}$.

Since in the pooling problem, we send flows from inputs to outputs via pools, we can create a super-sink node that connects to all outputs and consider each input $i \in I$ to be a unique commodity. The flow of commodity i on arc (l, j) is given by $v_{ilj} = q_{il}y_{lj}$ for $l \in L, j \in L \cup J, i \in I_l$. In order to ensure flow balance of commodity i at pool l, we add the constraints

$$y_{il} + \sum_{\substack{l' \in L: \\ i \in I_{l'}}} q_{il'} y_{l'l} = q_{il} \sum_{j \in L \cup J} y_{lj}, \quad l \in L, i \in I_l. \tag{8}$$

In the context of the p-formulation, specifications can be interpreted as commodities and (4a) serves the role of commodity balance constraints. Equations (7) and (8) make the flow balance constraints (1) redundant.

The specification requirement constraints at the output are modeled as

$$\sum_{i \in I} \lambda_{ik} y_{ij} + \sum_{l \in L} \sum_{i \in I_l} \lambda_{ik} q_{il} y_{lj} \leq \mu_{jk}^{\max} \sum_{i \in I \cup L} y_{ij}, \quad j \in J, k \in K, \tag{9a}$$

$$\sum_{i \in I} \lambda_{ik} y_{ij} + \sum_{l \in L} \sum_{i \in I_l} \lambda_{ik} q_{il} y_{lj} \geq \mu_{jk}^{\min} \sum_{i \in I \cup L} y_{ij}, \quad j \in J, k \in K. \tag{9b}$$

The q-formulation for pooling problem can now be stated as follows.

$$\begin{aligned} \min_{y,p} \quad & \sum_{(i,j)\in\mathcal{A}} c_{ij} y_{ij} \\ \text{s.t.} \quad & y \in \mathcal{F}, \\ & (7) - (9). \end{aligned} \tag{$\mathbb{Q}$}$$

It is easily observed that in the case of standard pooling problems, the above formulation reduces to the one proposed by Ben-Tal et al. [25].

1.2.3.2 pq-formulation

The pq formulation, introduced by Tawarmalani and Sahinidis [101] for standard pooling problems, is obtained by appending some valid inequalities to the q-formulation. These inequalities are given by

$$\sum_{i\in I_l} q_{il} y_{lj} = y_{lj}, \quad l \in L, j \in L \cup J, \tag{10a}$$

$$\sum_{j\in L\cup J} q_{il} y_{lj} \leq C_l q_{il}, \quad l \in L, i \in I_l. \tag{10b}$$

derived via the Reformulation Linearization Technique (RLT) [97] by multiplying (7) with y_{lj} and (2b) with q_{il}. These constraints were independently derived by Quesada and Grossmann [85] for processing network problems. The resulting formulation is:

$$\begin{aligned} \min_{y,p} \quad & \sum_{(i,j)\in\mathcal{A}} c_{ij} y_{ij} \\ \text{s.t.} \quad & y \in \mathcal{F}, \\ & (7) - (10). \end{aligned} \tag{$\mathbb{PQ}$}$$

The addition of (10) helps to obtain a significantly stronger polyhedral relaxation of the pooling problem, as explained in §1.5.2.1.

1.2.3.3 A hybrid formulation

Audet et al. [14] suggested a model that combined the p and q variables along with the y variables. The motivation was to avoid having bilinear terms of the form $q_{il}q_{jl'}$ that would arise by a straightforward extension of the Ben-Tal et al. model to the case of generalized

pooling problems. In this so-called hybrid model, proportion variables are used for the pools in L_I, i.e. pools with incoming arcs from some input nodes, and concentration variables are used for pools in $L \setminus L_I$. Let this hybrid formulation be denoted by ($\mathbb{HYB}$).

$$\min_{y,p} \quad \sum_{(i,j)\in\mathcal{A}} c_{ij} y_{ij}$$

$$\text{s.t.} \quad y \in \mathcal{F}, \quad (6) \quad \text{for } l \in L \setminus L_I$$

$$(7) \quad \text{for } l \in L_I, \qquad y_{il} = q_{il} \sum_{j\in L\cup J} y_{lj}, \quad l \in L_I, i \in I_l,$$

$$(10a) \quad \text{for } l \in L_I, j \in L \cup J, \qquad (10b) \quad l \in L_I, i \in I_l,$$

$$\sum_{i\in I} \lambda_{ik} y_{il} + \sum_{l'\in L_I} \sum_{i\in I_{l'}} \lambda_{ik} q_{il'} y_{l'l} + \sum_{l'\in L\setminus L_I} p_{l'k} y_{l'l} = p_{lk} \sum_{j\in L\cup J} y_{lj}, \quad l \in L \setminus L_I, k \in K,$$

$$\sum_{i\in I} \lambda_{ik} y_{ij} + \sum_{l\in L_I} \sum_{i\in I_l} \lambda_{ik} q_{il} y_{lj} + \sum_{l\in L\setminus L_I} p_{lk} y_{lj} \le \mu_{jk}^{\max} \sum_{i\in I\cup L} y_{ij}, \quad j \in J, k \in K,$$

$$\sum_{i\in I} \lambda_{ik} y_{ij} + \sum_{l\in L_I} \sum_{i\in I_l} \lambda_{ik} q_{il} y_{lj} + \sum_{l\in L\setminus L_I} p_{lk} y_{lj} \ge \mu_{jk}^{\min} \sum_{i\in I\cup L} y_{ij}, \quad j \in J, k \in K.$$

($\mathbb{HYB}$)

1.2.3.4 Equivalence of formulations

We now formally prove the correctness of the forgoing formulations for the pooling problem on acyclic networks. Two formulations are said to be equivalent if they have the same objective function and for every feasible point in one formulation, there exists a feasible point in the other formulation and vice versa.

Proposition 1.1. *Formulations* ($\mathbb{P}$), ($\mathbb{Q}$), ($\mathbb{PQ}$), *and* ($\mathbb{HYB}$) *are equivalent.*

Proof. First let us show that for any feasible point (q, y) in ($\mathbb{Q}$) there exists some p satisfying (4a) and (5). In particular, this p is given by $p_{lk} = \sum_{i\in I_l:\ (i,l)\in\mathcal{A}} \lambda_{ik} q_{il}, l \in L, k \in K$. Note that for any $l \in L$ and $l' \in L$ such that $(l', l) \in \mathcal{A}$, we have $I_{l'} \subseteq I_l$ and hence $I_{l'} \cap I_l = I_{l'}$. Choose a $k \in K$ and multiply both sides of equation (8) with λ_{ik}. Summing over $i \in I_l$

produces

$$\begin{aligned}\sum_{i\in I_l}\lambda_{ik}y_{il}+\sum_{l'\in L}\sum_{i\in I_{l'}\cap I_l}\lambda_{ik}q_{il}^{'}\,y_{l'l} &= \sum_{i\in I_l}\lambda_{ik}q_{il}\sum_{j\in L\cup J}y_{lj}\\ \Rightarrow\sum_{i\in I}\lambda_{ik}y_{il}+\sum_{l'\in L}\sum_{i\in I_{l'}}\lambda_{ik}q_{il}^{'}\,y_{l'l} &= \sum_{i\in I_l}\lambda_{ik}q_{il}\sum_{i\in I\cup L}y_{il}\end{aligned}$$

Setting $p_{lk}=\sum_{i\in I_l}\lambda_{ik}q_{il}$ for $l\in L,k\in K$ gives a feasible point (p,y) in $(\mathbb{P})$.

Now let us show that for any feasible point (p,y) in $(\mathbb{P})$ there exists some q satisfying (7), (8), and $p_{lk}=\sum_{i\in I_l}\lambda_{ik}q_{il}$ for $l\in L,k\in K$. For $l\in L,j\in L\cup J$, define ξ_{lj} to be the fraction of outgoing flow from l directed towards j,

$$\xi_{lj}:=\frac{y_{lj}}{\sum_{j\in L\cup J}y_{lj}}$$

Let T_{il} be the set of directed paths between $i\in I_l$ and l. Take a directed path $\tau:=\{i,\tau_1,\ldots,\tau_{m(\tau)},l\}\in T_{il}$. Since G is acyclic, there are no directed cycles on this path. Then the total flow from i that reaches l along τ is

$$\sigma_{il}^{\tau}=y_{i\tau_1}\xi_{\tau_{m(\tau)}l}\prod_{o=1}^{m(\tau)-1}\xi_{\tau_o\tau_{o+1}}.$$

Construct the q variables as follows

$$q_{il}=\frac{\sum_{\tau\in T_{il}}\sigma_{il}^{\tau}}{\sum_{i\in I\cup L}y_{il}}$$

The flow balance equations (1) imply that there is no supply at pools and all the supply originates at inputs. Hence, the quantity $\sum_{i\in I_l}\sum_{\tau\in T_{il}}\sigma_{il}^{\tau}$, which designates the total flow from all inputs to l must equal the total flow into l which is $\sum_{i\in I\cup L}y_{il}$. Hence $\sum_{i\in I_l}q_{il}=1$. Similarly, the total quanity of spec k at pool l is given by $\sum_{i\in I_l}\lambda_{ik}\sum_{\tau\in T_{il}}\sigma_{il}^{\tau}$ and hence $p_{lk}=\sum_{i\in I_l}\lambda_{ik}q_{il}$. Now the left hand side of (8) is

$$\begin{aligned}&y_{il}+\sum_{\substack{l'\in L:\\ i\in I_{l'}}}\sum_{\tau\in T_{il'}}\sigma_{il'}^{\tau}\frac{y_{l'l}}{\sum_{i\in I\cup L}y_{il'}}\\ =\;&y_{il}+\sum_{\substack{l'\in L:\\ i\in I_{l'}}}\sum_{\tau\in T_{il'}}\sigma_{il'}^{\tau}\xi_{l'l}\\ =\;&\sum_{\tau\in T_{il}}\sigma_{il}^{\tau}\\ =\;&q_{il}\sum_{j\in L\cup J}y_{lj}.\end{aligned}$$

Thus, we have shown that ($\mathbb{P}$) and ($\mathbb{Q}$) are equivalent formulations of the pooling problem. The equivalence of ($\mathbb{Q}$) and ($\mathbb{PQ}$) follows by noting that ($\mathbb{PQ}$) is obtained by appending valid inequalities (10) to ($\mathbb{Q}$). Finally, the correctness of ($\mathbb{HYB}$) can be shown using the steps of the above proof for pools in L_I. □

1.3 Problem sizes

The alternate formulations of §1.2.3 present different ways of modeling the p-formulation of the pooling problem obtained from Definition 1.1. All these equivalent formulations use the same flow variables on the arc set $\mathcal{A}$ and they only differ in the use of non-flow variables and additional constraints. Since bilinearities are what makes the pooling problem hard to solve, we mention the number of bilinear terms and bilinear constraints along with the number of non-flow variables in Table 1.

Table 1: Comparing problem sizes for alternate formulations of the pooling problem.

Formulation	Non-flow variables	Bilinear terms	Bilinear constraints	
			Equality	Inequality
$\mathbb{P}$	$\|K\|\|L\|$	$\|K\|\|L\|\,\mathcal{O}(\|L\|+\|J\|)$	$\|K\|\|L\|$	$2\|K\|\|J\|$
$\mathbb{Q}$	$\sum_{l\in L}\|I_l\|$	$\sum_{l\in L}\|I_l\|\,\mathcal{O}(\|L\|+\|J\|)$	$\sum_{l\in L}\|I_l\|$	$2\|K\|\|J\|$
$\mathbb{PQ}$	$\sum_{l\in L}\|I_l\|$	$\sum_{l\in L}\|I_l\|\,\mathcal{O}(\|L\|+\|J\|)$	$\sum_{l\in L}\|I_l\| +$ $\|L\|\,\mathcal{O}(\|L\|+\|J\|)$	$2\|K\|\|J\|$ $+\sum_{l\in L}\|I_l\|$
$\mathbb{HYB}$	$\sum_{l\in L_I}\|I_l\|$ $+\|K\|\|L\setminus L_I\|$	$\Big[\sum_{l\in L_I}\|I_l\| + \|K\|\|L\setminus L_I\|\Big]$ $\times\mathcal{O}(\|L\setminus L_I\|+\|J\|)$	$\sum_{l\in L_I}\|I_l\| +$ $\|K\|\|L\setminus L_I\| +$ $\|L_I\|\,\mathcal{O}(\|L\setminus L_I\|+\|J\|)$	$2\|K\|\|J\|$ $+\sum_{l\in L_I}\|I_l\|$

The table suggests that when $|K| << |I|$ and G is dense, then ($\mathbb{P}$) will have fewer variables, bilinear terms, and bilinear equalities than ($\mathbb{Q}$). The smaller size of ($\mathbb{P}$) in this particular case may prove to be advantageous while solving the pooling problem to global optimality.

1.4 Variants of the pooling problem

We have already mentioned two types of pooling problems - standard and generalized, depending on the absence or presence of arcs between pools, respectively. A broader class of network flow problems with bilinear terms is described by Lee and Grossmann [66], Quesada and Grossmann [85]. Nonlinear blending rules have also been proposed, see Misener and Floudas [74] for a discussion. One such example of nonlinear blending where the bilinear terms in the pooling problem are replaced by cubic terms was recently proposed by Realff et al. [86]. Ruiz et al. [92] studied a variant of the standard pooling problem where total flow into an output, given by $\sum_{i \in I \cup L} y_{ij}$, is fixed to some positive constant, for each output $j \in J$. Under this assumption, Ruiz et al. scaled the output requirement constraints (5) with the total flow $\sum_{i \in I \cup L} y_{ij}$ and proposed a formulation with bilinear objective function, bilinear inequalities, and linear constraints using only ratio variables for inflows and outflows from each node.

An extended pooling problem that imposes upper bounds on emissions from outputs, based on regulations set by the Environmental Protection Agency (EPA), was introduced in Misener et al. [77]. The EPA model was developed as a mixed integer nonlinear program (MINLP) by Furman and Androulakis [46], thus making the extended pooling problem also a MINLP. Other examples of MINLP models can be found in the works of D'Ambrosio et al. [36], Meyer and Floudas [72], Misener and Floudas [75], Nishi [80], Visweswaran [110]. These MINLP variants arise mainly by including binary decision variables related to the use of each arc or node in the graph or forcing the flows to be semicontinuous. The wastewater treatment problem of Karuppiah and Grossmann [64] is another MINLP that is closely related to the pooling problem.

It is worth mentioning here that since the p-formulation is the most natural way of modeling the pooling problem stated in Definition 1.1, one can easily obtain a p-formulation for each of the variants. However, not all variants admit a tractable q-formulation. For example, the model proposed by Meyer and Floudas [72] includes a removal ratio parameter

δ_{lk} of spec k at pool l which appears in the spec tracking constraint as

$$\sum_{i \in I} \lambda_{ik} y_{il} + \sum_{l' \in L} p_{l'k} y_{l'l} = (1 - \delta_{lk})\, p_{lk} \sum_{j \in L \cup J} y_{lj}, \quad l \in L, k \in K.$$

In the construction of $(\mathbb{Q})$, we introduced proportion variables q_{il} that were independent of the path between i and l. However, in the presence of the parameter δ_{lk}, this is not possible since we have to account for the fractional loss $\delta_{l'k}$ incurred at each intermediate node l' in a path from i to l. Hence it becomes necessary to introduce path-dependent proportion variables, which can lead to a prohibitive increase in the size of the q-formulation for this particular variant.

Finally, we propose a new variant of the pooling problem that includes discrete decisions and a planning horizon for demand to be met at the outputs. Such problems may arise as a substructure in maritime inventory scheduling operations; see for example Al-Khayyal and Hwang [4].

1.4.1 Time indexed pooling problem

Consider a generalized pooling problem and let T be a set of time periods. For each time period $t \in T$, we have to make the following decisions: 1) semicontinuous flow y_{ijt} on arc $(i, j) \in \mathcal{A}$, 2) s_{it} amounts of inventory to be held at a node $i \in \mathcal{N}$, 3) $x^{in}_{lt} = 1$ iff there is inflow at pool l, 4) $x^{out}_{lt} = 1$ iff there is outflow at pool l, 5) $z_{lt} = 1$ iff pool l is used for mixing.

Some additional parameters are required for this model. Let a_{it} and d_{jt} be the supply at input $i \in I$ and demand at output $j \in J$, respectively, at time $t \in T$. Let h_l be the fixed cost of using a pool $l \in L$. The set of pools is partitioned into two categories - L_c and $L \setminus L_c$. A pool $l \in L_c$ is allowed to be leased on a contract basis for a fixed period τ_l and can only be used under contract. Typically, $\tau_l \leq |T|$ and the contracts are renewable. For a pool $l \in L_c$, the fixed cost h_l is associated with the entire contract.

We first state the p-formulation of this problem. p_{lkt} denotes the concentration value of spec k at pool l at time t.

$$
\begin{aligned}
\min \quad & \sum_{t\in T}\sum_{(i,j)\in A} c_{ij}y_{ijt} \; + \sum_{t\in T}\sum_{l\in L} h_l z_{lt}, \\
& a_{it} + s_{i(t-1)} = \textstyle\sum_{l\in L\cup J} y_{ilt} + s_{it}, \quad i \in I, t \in T, \\
& \sum_{i\in I\cup L} y_{ilt} + s_{l(t-1)} = s_{lt} + \sum_{j\in L\cup J} y_{ljt}, \quad l \in L, t \in T, \\
& \sum_{l\in I\cup L} y_{ljt} + s_{j(t-1)} = s_{jt} + d_{jt}, \quad j \in J, t \in T, \\
& \sum_{i\in I} \lambda_{ik} y_{ilt} + \sum_{l'\in L} p_{l'kt} y_{l'lt} + p_{lk(t-1)} s_{l(t-1)}, \\
& \qquad\qquad = p_{lkt}\left[\sum_{j\in L\cup J} y_{ljt} + s_{lt}\right], \quad l \in L, k \in K, t \in T, \\
& \sum_{i\in I} \lambda_{ik} y_{ijt} + \sum_{l\in L} p_{lkt} y_{ljt} \le \mu_{jk}^{\max} \sum_{i\in I\cup L} y_{ijt}, \quad j \in J, k \in K, t \in T, \\
& \sum_{i\in I} \lambda_{ik} y_{ijt} + \sum_{l\in L} p_{lkt} y_{ljt} \ge \mu_{jk}^{\min} \sum_{i\in I\cup L} y_{ijt}, \quad j \in J, k \in K, t \in T, \\
& (y, s, x^{in}, x^{out}, z) \in \mathcal{Z}, \\
& 0 \le s_{lt} \le C_l, \; l \in L, t \in T,
\end{aligned}
\tag{($\mathbb{P}$-Inv)}
$$

where $\mathcal{Z}$ represents the set of combinatorial constraints that make this optimization model a mixed integer bilinear program (MIBLP).

$$
\begin{aligned}
\mathcal{Z} := \Big\{ (y, s, x^{in}, x^{out}, z) : \\
& y_{ijt} \in \{0\} \cup [\ell_{ij}, u_{ij}], \; (i,j) \in \mathcal{A}, t \in T, x_{lt}^{in}, x_{lt}^{out}, z_{lt} \in \{0,1\}, \; l \in L, t \in T, && \text{(11a)} \\
& x_{lt}^{in} + x_{lt}^{out} \le z_{lt}, \qquad l \in L \setminus L_c, t \in T, && \text{(11b)} \\
& x_{lt}^{in} + x_{lt}^{out} \le \min\Big\{1, \sum_{t'=t-\tau_l+1}^{t} z_{lt'}\Big\}, \qquad l \in L_c, t \in T, && \text{(11c)} \\
& y_{ilt} \le u_{il} x_{lt}^{in}, \quad l \in L, i \in I \cup L, t \in T, && \text{(11d)} \\
& y_{ljt} \le u_{lj} x_{lt}^{out}, \qquad l \in L, j \in L \cup J, t \in T, && \text{(11e)} \\
& s_{lt} \le C_l \sum_{t'=t-\tau_l+1}^{t+1} z_{lt'}, \qquad l \in L_c, t \in T \Big\}. && \text{(11f)}
\end{aligned}
$$

The combinatorial constraints can be explained as follows. (11a) states variable definitions for semicontinuous flows and binary variables. Here, $z_{lt} = 1$ for $l \in L_c$ implies that a new

contract for pool l was started at time t whereas $z_{lt} = 1$ for $l \in L \setminus L_c$ implies that pool l was used at time t. (11b) and (11c) enforce either inflow or outflow at each pool and for contract pools, the constraint that there should be no flow if the contract has expired. The next two constraints (11d) and (11e) ensure consistency between incoming and outgoing binary variables and incoming and outgoing flows at each pool. The last constraint (11f) clears inventory at a pool if its contract is not renewed.

In order to obtain a q-formulation, we claim that time indexing can be treated in the same manner as pool-pool arcs. Let $G^{'}$ be a new graph whose nodes are partitioned into inputs $I^{'}$, pools $L^{'}$, and outputs $J^{'}$. $I^{'}$ consists of $|I||T|$ nodes, one for each input-time pair $[i,t]$ for $i \in I, t \in T$. Similarly, $L^{'}$ and $J^{'}$ have $|L||T|$ and $|J||T|$ nodes, respectively. Although this proposed construction of $G^{'}$ includes input-input arcs and output-output arcs to model inventory at time t, these arcs can be easily eliminated by introducing auxiliary pools, one for each $[i,t]$ and one for each $[j,t]$. An auxiliary pool for node $[i,t]$ is directly connected to $[i,t]$ and $[i,t-1]$. Hence it has a direct arc from input i at time t and has paths from input i at time t', for all $t' < t$. Since the specification levels at any input are independent of time, auxiliary pool i possesses λ_{ik} level of specification k for all time periods t. Hence, we do not need ratio variables for auxiliary pools connected to inputs. An auxiliary pool connected to output j at time t has two outgoing arcs: one to $[j,t]$ and another to $[j,t+1]$. Since the specification requirement constraints at output j at time $t+1$ do not depend on specification of inventory stored from time t at j, we do not need ratio variables for these auxiliary pools. Now consider a node $[l,t] \in L^{'}$. We need ratio variables for each such node. The the set of inputs in $G^{'}$ from which there exists a directed path to $[l,t]$ is given by $I^{'}_{[l,t]} = \{[i,t^{'}] \in I^{'} : i \in I_l, t^{'} \leq t\}$, i.e. all the input nodes in I that had a path to l and time index before t. Thus the proportion variable $q_{ilt^{'}t}$ denotes the fraction of incoming flow at pool l at time t which is contributed by input $i \in I$ from time $t^{'} \leq t$. For any outflow arc $(l,j) \in \mathcal{A}$ from pool l, we have the bilinear terms $v_{iljt^{'}t} = q_{ilt^{'}t}y_{ljt}$ and $v^{s}_{ilt^{'}t} = q_{ilt^{'}t}s_{lt}$. We can now formulate the time indexed pooling problem on $G^{'}$ using the q- or pq-formulation for generalized pooling problems.

1.5 Relaxations

The pooling problem is a nonconvex problem where nonconvexities arise due to the presence of bilinear terms. For the two formulations ($\mathbb{P}$) and ($\mathbb{PQ}$), bilinearities are present in equations (4a), (5) and (8), (9), respectively. We observe that all the bilinear terms in these two formulations are of the form $w_{lkj} = p_{lk}y_{lj}$ and $v_{ilj} = q_{il}y_{lj}$, respectively. Hence the bilinear terms arise for each pool l, each outgoing arc j from this pool l, and each commodity (specification $k \in K$ for ($\mathbb{P}$) and input $i \in I_l$ for ($\mathbb{PQ}$)) arriving at this pool l. In this section, we analyze properties of the commonly used relaxations for the pooling problem.

1.5.1 Envelopes of bilinear functions

One way of relaxing a nonconvex function is to obtain the convex underestimator and concave overestimator of the function over its domain. When the function consists of a single bilinear term, we wish to relax the set

$$\mathcal{T} := \{(\chi, \rho, \omega) \in \Re \times \Re \times \Re \colon \omega = \chi\rho,\ \chi \in [a_1, b_1],\ \rho \in [a_2, b_2]\}. \tag{12}$$

McCormick [70] proposed the following four inequalities to relax $\mathcal{T}$

$$\omega \geq b_2\chi + b_1\rho - b_1b_2, \qquad \omega \geq a_2\chi + a_1\rho - a_1a_2, \tag{13a}$$

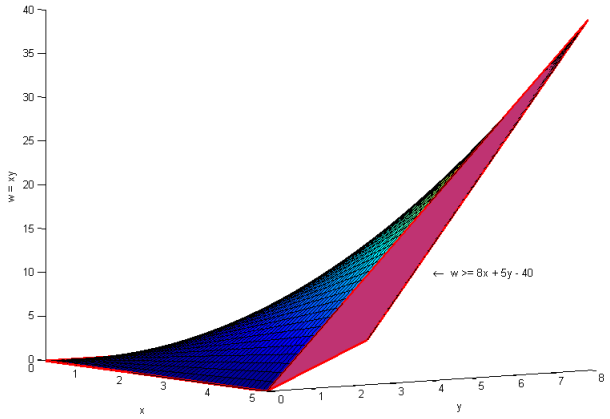
$$\omega \leq b_2\chi + a_1\rho - a_1b_2, \qquad \omega \leq a_2\chi + b_1\rho - a_2b_1. \tag{13b}$$

Here (13a) and (13b) define the convex and concave envelope of $\omega = \chi\rho$ over the rectangle $[a_1, a_2] \times [b_1, b_2]$, respectively, and are commonly referred to as the *McCormick envelopes*. Later, Al-Khayyal and Falk [5] proved that the inequalities of (13) in fact define $\text{conv}(\mathcal{T})$. For brevity, we will denote the McCormick relaxation (13) by either $(\chi, \rho, \omega) \in \mathcal{M}(\mathcal{T})$ or $(\chi, \rho, \omega) \in \mathcal{M}(\{\omega = \chi\rho\})$, where the bounding box $[a_1, b_1] \times [a_2, b_2]$ will be the natural bounds on the associated variables unless otherwise stated explicitly. These envelopes are depicted in Figure 2. One can also observe that envelopes (13) can be obtained from the following four valid multiplications,

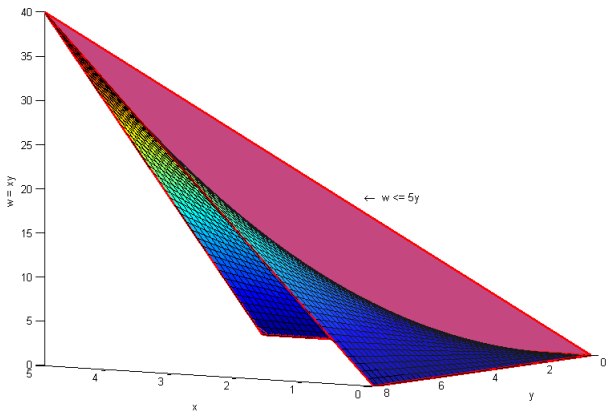
$$(\chi - b_1)(\rho - b_2) \geq 0, \qquad (\chi - a_1)(\rho - a_2) \geq 0$$

$$(\chi - a_1)(\rho - b_2) \leq 0, \qquad (\chi - b_1)(\rho - a_2) \leq 0$$

upon substituting $\omega = \chi\rho$. Such variable bound factor multiplication is a basic principle of the Reformulation Linearization Technique (RLT) of Sherali and Adams [97] for building relaxations of mixed discrete and nonconvex sets. Hence, McCormick envelopes are also RLT constraints.



(a) Convex envelope $\omega \geq 8\chi + 5\rho - 40$. Hidden facet is $\omega \geq 0$.



(b) Concave envelope $\omega \leq 5\rho$. The other facet is $\omega \leq 8\chi$.

Figure 2: McCormick relaxation for the set $\{(\chi, \rho, \omega) \colon \omega = \chi\rho, \chi \in [0, 5], \rho \in [0, 8]\}$.

The single term McCormick envelopes of (13) are a common choice for relaxations used in a branch-and-bound algorithm for solving pooling problems, perhaps dating back to Foulds et al. [43]. Thus one can create polyhedral relaxations of the pooling problem by replacing every occurrence of a bilinear term $\chi\rho$ with a new variable ω and adding the

four inequalities from (13). This procedure can be applied to any of the proposed problem formulations. In §1.5.2.1, we shall formally compare the strengths of McCormick relaxations of the various formulations. First let us state these polyhedral relaxations of the pooling problem.

We consider only the two formulations ($\mathbb{P}$) and ($\mathbb{PQ}$). The relaxation of ($\mathbb{PQ}$) is stronger than ($\mathbb{Q}$) since ($\mathbb{PQ}$) contains additional valid inequalities (10). The relaxation of ($\mathbb{HYB}$) can be obtained analogously from that of ($\mathbb{P}$) and ($\mathbb{PQ}$). For ($\mathbb{P}$), we introduce an auxiliary variable w_{lkj} to represent $p_{lk}y_{lj}$ and relax this bilinear term using (13).

$$
\begin{aligned}
\mathcal{M}(\mathbb{P}) := \Big\{ (p,y,w) \colon\ & y \in \mathcal{F},\ \underline{p}_{lk} \le p_{lk} \le \bar{p}_{lk},\ l \in L, k \in K \\
& \sum_{i \in I} \lambda_{ik} y_{il} + \sum_{l' \in L} w_{l'kl} = \sum_{j \in L \cup J} w_{lkj}, \quad l \in L, k \in K \\
& \sum_{i \in I} \lambda_{ik} y_{ij} + \sum_{l \in L} w_{lkj} \le \mu_{jk}^{\max} \sum_{i \in I \cup L} y_{ij}, \quad j \in J, k \in K \qquad \text{(p-relax)} \\
& \sum_{i \in I} \lambda_{ik} y_{ij} + \sum_{l \in L} w_{lkj} \ge \mu_{jk}^{\min} \sum_{i \in I \cup L} y_{ij}, \quad j \in J, k \in K \\
& (p_{lk}, y_{lj}, w_{lkj}) \in \mathcal{M}(\{w_{lkj} = p_{lk} y_{lj}\}), \quad l \in L, j \in L \cup J, k \in K \Big\}.
\end{aligned}
$$

For ($\mathbb{PQ}$), let $v_{ilj} = q_{il}y_{lj}$, for $l \in L, i \in I_l, j \in L \cup J$, and relax this bilinear term using (13).

$$
\begin{aligned}
\mathcal{M}(\mathbb{PQ}) := \Big\{ (q,y,v) \colon\ & y \in \mathcal{F},\ q_l \in \Delta^{|I_l|},\ l \in L \\
& y_{il} + \sum_{\substack{l' \in L: \\ i \in I_{l'}}} v_{il'l} = \sum_{j \in L \cup J} v_{ilj}, \quad l \in L, i \in I_l \\
& \sum_{i \in I} \lambda_{ik} y_{ij} + \sum_{l \in L} \sum_{i \in I_l} \lambda_{ik} v_{ilj} \leq \mu^{\max}_{jk} \sum_{i \in I \cup L} y_{ij}, \quad j \in J, k \in K \\
& \sum_{i \in I} \lambda_{ik} y_{ij} + \sum_{l \in L} \sum_{i \in I_l} \lambda_{ik} v_{ilj} \geq \mu^{\min}_{jk} \sum_{i \in I \cup L} y_{ij}, \quad j \in J, k \in K \qquad \text{(pq-relax)} \\
& \sum_{i \in I_l} v_{ilj} = y_{lj}, \quad l \in L, j \in L \cup J \\
& \sum_{j \in L \cup J} v_{ilj} \leq C_l q_{il}, \quad l \in L, i \in I_l \\
& (q_{il}, y_{lj}, v_{ilj}) \in \mathcal{M}(\{v_{ilj} = q_{il} y_{lj}\}), \quad l \in L, i \in I_l, j \in L \cup J \Big\}.
\end{aligned}
$$

It is reasonable to consider relaxing not just the individual bilinear terms, but also the entire bilinear function that appears in a constraint, for e.g. $\sum_{l \in L} p_{lk} y_{lj}$ in (5). For an arbitrary nonconvex function, stronger relaxations can be obtained by developing under- and over-estimators for the entire function (cf. Tawarmalani and Sahinidis [102]). When this function consists of sums of multiple bilinear terms, the convex and concave envelopes are known to be polyhedral due to a result on multilinear functions by Rikun [89], and later also by Sherali [96]. In general, it is not true that sum of convex (concave) envelopes of individual functions gives the convex (concave) envelope of the sum of functions. Meyer and Floudas [71], Rikun [89], Tardella [100] develop some sufficient conditions when this holds true. However partly because of the presence of some special structure, the pooling problem always admits the sum decomposition rule for the bilinear functions arising in it. We first define this structure.

Definition 1.2. A bilinear function is said to be *bipartite* if its co-occurence graph, [cf. 55], whose nodes correspond to variables and edges correspond to bilinear terms, is bipartite.

Since all the bilinear terms in the pooling problem are of the form $w_{lkj} = p_{lk} y_{lj}$ or $v_{ilj} = q_{il} y_{lj}$, it follows that all the bilinear functions are bipartite. In general, the bipartite

condition is not sufficient to guarantee the tightness of the individual McCormick envelopes. However, in our case, it is indeed true, as observed next.

Observation 1.2. *For the pooling problem, envelopes of bilinear functions taken over bounds on the associated variables are given by single term McCormick inequalities* (13).

Proof. It suffices to concentrate on formulation (ℙ). Similar reasonings follow for the alternate formulations. In (ℙ), the bilinear functions are $\sum_{l\in L} p_{lk}y_{lj}$ in (5) and $\sum_{l'\in L} p_{l'k}y_{l'l}$ and $p_{lk}\sum_{j\in L\cup J} y_{lj}$ in (4). The envelope of each of these functions is sum decomposable because the first and second functions are separable [39] and the third function is bipartite with positive coefficients [69, Theorem 6]. Consider equation (4a) and the combined function $\sum_{l'\in L} p_{l'k}y_{l'l} - p_{lk}\sum_{j\in L\cup J} y_{lj}$. This is a bipartite bilinear function with both positive and negative coefficients. A direct application of [69, Theorem 9] implies that the McCormick relaxation of this function is weaker than its envelopes. However, since p_{lk} and $p_{l'k}$ have some finite bounds, say $[0, 1]$, translating p_{lk} and $p_{l'k}$ as $1 - \bar{p}_{lk}$ and $\bar{p}_{l'k} + 1$, respectively, gives a transformed function $\sum_{l'\in L} \bar{p}_{l'k}y_{l'l} + \bar{p}_{lk}\sum_{j\in L\cup J} y_{lj}$. Now the result of [69, Theorem 8] implies that the envelopes of this transformed function are given by all the McCormick envelopes of individual bilinear terms. □

Remark 1.1. We would like to note here that the statement of Observation 1.2 is slightly stronger than that of Misener and Floudas [76], Property 3.1.3.1, in the sense that it also addresses the bilinear function appearing in (4a) that was modified using flow balance constraint (1) at pool l.

1.5.1.1 Piecewise linear

The strength of the McCormick envelopes (13) for a single bilinear term, given by $\mathcal{T}$, depends on the bounds $[a_1, b_1]$ and $[a_2, b_2]$ on the variables χ and ρ, respectively. Tighter bounds lead to stronger relaxations. Hence, partitioning the intervals of one or both the variables and then constructing McCormick envelopes in each interval gives a much stronger relaxation than simply including equations (13) based on the entire interval. Of course, the level of partitioning determines the strength of this new relaxation. To enforce validity

of this relaxation, one must add extra binary variables to turn on/off each partition with exactly one partition being turned on. This gives rise to a MILP relaxation, referred to as the piecewise linear McCormick relaxation, of the set $\mathcal{T}$.

Piecewise linear McCormick relaxations were used by Meyer and Floudas [72] to solve some generalized pooling problems. Hasan and Karimi [58], Wicaksono and Karimi [113] proposed alternative MILP models for piecewise linear relaxations. An extensive computational study on small scale pooling problems was performed by Gounaris et al. [50] to investigate different partitioning levels and MILP models. Recently, Misener et al. [78] implemented a generic branch-and-bound based solver for pooling problems that uses piecewise linear MILP relaxations at each node of the branch-and-bound tree.

1.5.2 Relaxing feasible sets

Next we turn our attention to finding good relaxations for constraints in the pooling problem. First, we are interested in studying a relaxation of the feasible set that arises at each pool. Let $\mathcal{F}_l$ be the set of feasible outgoing flows from a pool l.

$$\mathcal{F}_l = \left\{ y_{l\cdot} : \sum_{j \in L \cup J} y_{lj} \leq C_l,\ 0 \leq y_{lj} \leq u_{lj}, j \in L \cup J \right\}, \tag{14}$$

where $y_{l\cdot} = \{y_{lj}\}_{j \in L \cup J}$ is the vector of outgoing flow variables from l. We also denote $p_l = \{p_{lk}\}_{k \in K}$ and $q_l = \{q_{il}\}_{i \in I_l}$ as the vectors of unknown specifications and incoming flow ratio from inputs at a pool l, respectively. For the two formulations $(\mathbb{P})$ and $(\mathbb{Q})$, relaxations of the feasible sets corresponding to pool l are given by $\mathcal{P}_l$ and $\mathcal{Q}_l$, respectively, defined as

$$\begin{aligned} \mathcal{P}_l := \Big\{ \Big(p_l, y_{l\cdot}, \{w_{lkj}\}_{\substack{k \in K \\ j \in L \cup J}} \Big) : & \ w_{lkj} = p_{lk} y_{lj},\ k \in K, j \in L \cup J \\ & \underline{p}_{lk} \leq p_{lk} \leq \bar{p}_{lk}, k \in K, \quad y_{l\cdot} \in \mathcal{F}_l \Big\}. \end{aligned} \tag{15a}$$

$$\begin{aligned} \mathcal{Q}_l := \Big\{ \Big(q_l, y_{l\cdot}, \{v_{ilj}\}_{\substack{i \in I_l \\ j \in L \cup J}} \Big) : & \ v_{ilj} = q_{il} y_{lj},\ i \in I_l, j \in L \cup J \\ & q_l \in \Delta^{|I_l|}, \quad y_{l\cdot} \in \mathcal{F}_l \Big\}. \end{aligned} \tag{15b}$$

The above single pool relaxations are constructed by dropping the incoming arcs at pool l along with their respective bounds and the commodity balance constraints (4a) and (8) for $(\mathbb{P})$ and $(\mathbb{Q})$, respectively. Observe that we have also included new variables w_{lkj} and v_{ilj} for the bilinear terms in $\mathcal{P}_l$ and $\mathcal{Q}_l$, respectively.

1.5.2.1 Comparing p- and pq-relaxations

Our main purpose in this section is to prove that $\mathcal{M}(\mathbb{PQ})$ admits a stronger polyhedral relaxation of the pooling problem than $\mathcal{M}(\mathbb{P})$. We state this result in Proposition 1.3. Towards this end, consider the relaxation $\mathcal{M}(\mathbb{PQ})$. It contains linear inequalities and equalities, some of which are corresponding to each pool l. Hence, one may expect that studying single pool relaxations of a pooling problem can yield relaxations for the entire problem. Recall the set $\mathcal{Q}_l$ defined in (15b) as the single pool relaxation obtained by dropping the incoming arcs at pool l along with their respective bounds and the commodity balance constraints (8). We now convexify $\mathcal{Q}_l$ for every $l \in L$. Towards this end, we first prove a general result on bilinear terms. This result is a special case of Theorem 3.1 from Chapter 3. However, for the sake of self-containment, we prove it here independently.

A result on bilinear terms. Let $\mathcal{X}^+$ be a general bilinear set defined as follows

$$\mathcal{X}^+ := \{(\chi, \rho, \omega) \in \Re^m_+ \times \Re^n \times \Re^{m \times n} \colon \omega = \chi\rho^\top, \chi \in \Theta, \rho \in \Upsilon\}, \tag{16}$$

where $\Theta = \{\chi \in \Re^m_+ \colon \sigma^\top \chi = \sigma_0\}$ with $\sigma > \mathbf{0}, \sigma_0 > 0$ is a $(m-1)$-dimensional simplex in $\Re^m$ and Υ is some polytope in $\Re^n$.

Lemma 1.1. *Suppose that Υ is a polytope given as $\Upsilon = \{\rho \colon \Pi\rho \geq \pi_0\}$. Let $\omega_{i\cdot}$ denote the i^{th} row of ω. Then, the convex hull of $\mathcal{X}^+$ is*

$$\begin{aligned} \operatorname{conv}(\mathcal{X}^+) = \Big\{(\chi, \rho, \omega) \colon \quad \Pi\omega_{i\cdot}^\top \quad &\geq \pi_0\chi_i, \quad i = 1, \ldots, m \\ \sum_{i=1}^m \sigma_i \omega_{i\cdot}^\top \quad &= \sigma_0\rho, \;\; \chi \in \Theta\Big\}. \end{aligned}$$

Proof. Clearly the following is true.

$$\chi = \sum_{i=1}^m \chi_i \mathbf{e}_i = \sum_{i=1}^m \frac{\chi_i \sigma_i}{\sigma_0} \frac{\sigma_0 \mathbf{e}_i}{\sigma_i}.$$

Let us denote the extreme points of Υ by $\operatorname{ext}\Upsilon = \{\rho^\tau\}_{\tau \in T}$. Now the extreme points of $\operatorname{conv}(\mathcal{X}^+)$ are

$$\operatorname{ext}\operatorname{conv}(\mathcal{X}^+) = \bigcup_{\substack{i \in [m] \\ \tau \in T}} \left\{(\chi, \rho, \omega) \colon \chi = \frac{\sigma_0 \mathbf{e}_i}{\sigma_i}, \; \rho = \rho^\tau, \; \omega = \frac{\sigma_0 \mathbf{e}_i}{\sigma_i} \rho^{\tau\top}\right\}.$$

Since $\text{conv}(\mathcal{X}^+)$ is compact, any point $(\chi, \rho, \omega) \in \text{conv}(\mathcal{X}^+)$ can be expressed as a convex combination of its extreme points. This implies that $\text{conv}(\mathcal{X}^+) = \text{conv}(\bigcup_{i=1}^m \Psi_i)$, where

$$\Psi_i = \left\{(\chi, \rho, \omega)\colon \chi = \frac{\sigma_0 \mathbf{e}_i}{\sigma_i},\ \rho \in \Upsilon,\ \omega = \frac{\sigma_0 \mathbf{e}_i}{\sigma_i}\rho^\top\right\}$$

is a polytope for all $i = 1, \ldots, m$. Notice that we have already convexified the extreme points of Υ in the definition of Ψ_i. The recession cones of all the Ψ_i's are empty. Hence $\text{conv}(\cup_i \Psi_i)$ is a closed set. Using Balas' result [16] on disjunctive programming, we obtain an extended formulation as

$$\begin{aligned}\text{conv}(\mathcal{X}^+) = \text{Proj}_{\chi,\rho,\omega}\Big\{(\{\chi^i, \rho^i, \omega^i\}_{i\in[m]}, \chi, \rho, \omega, \lambda)\colon \chi = \sum_i \chi^i,\ \rho = \sum_i \rho^i,\ \omega = \sum_i \omega^i \\ \Pi\rho^i \geq \pi_0\lambda_i,\ \chi^i = \frac{\sigma_0\mathbf{e}_i\lambda_i}{\sigma_i},\ \omega^i = \frac{\sigma_0\mathbf{e}_i}{\sigma_i}{\rho^i}^\top,\ \forall i \\ \lambda \in [0,1],\ \sum_i \lambda_i = 1\Big\}.\end{aligned}$$

In order to obtain the projection, note that $\chi_i = \sigma_0\lambda_i/\sigma_i$, for all i, and hence $\lambda_i = \sigma_i\chi_i/\sigma_0$. Also $\omega_{i\cdot} = \omega^i_{i\cdot} = \sigma_0{\rho^i}^\top/\sigma_i$. Hence $\rho^i = \sigma_i\omega^\top_{i\cdot}/\sigma_0$. Now $\rho = \sum_i \rho^i$ implies that $\rho = \sum_i \sigma_i\omega^\top_{i\cdot}/\sigma_0$. After making these substitutions we get the desired result. □

We now use Lemma 1.1 to construct the convex hull of $\mathcal{Q}_l$.

Proposition 1.2. *The convex hull of $\mathcal{Q}_l$ is given by*

$$\begin{aligned}\text{conv}(\mathcal{Q}_l) = \Big\{\Big(q_l, y_{l\cdot}, \{v_{ilj}\}_{\substack{i\in I_l \\ j\in L\cup J}}\Big)\colon q_l \in \Delta^{|I_l|} \\ \sum_{j\in L\cup J} v_{ilj} \leq C_l q_{il}, \quad i \in I_l \\ \sum_{i\in I_l} v_{ilj} = y_{lj}, \quad j \in L\cup J \\ 0 \leq v_{ilj} \leq u_{lj}q_{il}, \quad i \in I_l, j \in L\cup J\Big\}.\end{aligned} \tag{17}$$

Proof. Directly from Lemma 1.1 with $\chi = q_l, \rho = y_{l\cdot}, \omega = v_{\cdot l\cdot}$ and the sets $\Theta = \Delta^{|I_l|}, \Upsilon = \mathcal{F}_l$. □

The result of Proposition 1.2 can be suitably modified if the set of feasible outgoing flows $\mathcal{F}_l$ contains additional inequalities. We also observe that the description of $\text{conv}(\mathcal{Q}_l)$

requires two McCormick inequalities for each bilinear term $v_{ilj} = q_{il}y_{lj}$ and the two valid inequalities included in the *pq*-formulation of §1.2.3.2.

While defining $\mathcal{Q}_l$ in (15b), we dropped the variables y_{il} for $i \in I \cup L$. We could have retained these variables along with their bounds and still applied Lemma 1.1 to obtain a tighter relaxation than the one presented in (17). However, this stronger relaxation comes at a cost of introducing McCormick inequalities for new bilinear terms of the form $\hat{v}_{i'il} = q_{il}y_{i'l}$, for $i' \in I \cup L$, thus considerably increasing the size of this relaxation. Since the bilinear terms $v_{ilj} = q_{il}y_{lj}$ also appear in the spec requirement constraints (9), the additional variables v_{ilj} introduced in (17) are no more than those necessary.

Lemma 1.1 relies on the fact that Θ is a simplex in order to project out the auxiliary variables introduced by the extended formulation. The result does not carry through when Θ is an arbitrary polytope, since all the auxiliary variables cannot be eliminated in the projection step. This property has an important implication in our context. Consider the single pool relaxation $\mathcal{P}_l$ for the formulation ($\mathbb{P}$). For the set $\mathcal{P}_l$, the variable $\chi = \{p_{lk}\}_k$ and the set Θ is a hypercube given by the tight bounds $[\underline{p}_l, \bar{p}_l]$ on the specification produced at this pool. Hence Lemma 1.1 cannot be applied to $\mathcal{P}_l$. Nonetheless, the convex hull of $\mathcal{P}_l$ can be obtained using a sequential convexification/level-$|K|$ RLT procedure of Sherali and Adams [97]. The relaxation of $\mathcal{P}_l$ using the McCormick envelopes of $w_{lkj} = p_{lk}y_{lj}$, denoted as $\mathcal{M}(\mathcal{P}_l)$, is just one step of this RLT procedure and hence weaker than the convex hull of $\mathcal{P}_l$.

Observation 1.3. $\text{conv}(\mathcal{P}_l) \subset \mathcal{M}(\mathcal{P}_l)$.

We are now ready to prove the dominance of the *pq*-relaxation.

Proposition 1.3. *The pq-relaxation $\mathcal{M}(\mathbb{PQ})$ is a stronger relaxation than the p-relaxation $\mathcal{M}(\mathbb{P})$ in the sense that for any $c \in \Re^{|\mathcal{A}|}$,*

$$\begin{aligned}\eta^{\mathbb{PQ}} = \min\{c^\top y \colon (q,y,v) \in \mathcal{M}(\mathbb{PQ})\} &\geq \eta^{\mathbb{HYB}} = \min\{c^\top y \colon (p,q,y,w,v) \in \mathcal{M}(\mathbb{HYB})\} \\ &\geq \eta^{\mathbb{P}} = \min\{c^\top y \colon (p,y,w) \in \mathcal{M}(\mathbb{P})\}.\end{aligned}$$

Proof. As seen in the proof of Proposition 1.1, every point in $\mathcal{Q}_l$ can be mapped to a point in $\mathcal{P}_l$ using the linear mappings $p_{lk} = \sum_{i \in I_l} \lambda_{ik} q_{il}$ and $w_{lkj} = \sum_{i \in I_l} \lambda_{ik} v_{ilj}$. It follows

that $\text{conv}(\mathcal{Q}_l)$ can be linearly mapped to $\text{conv}(\mathcal{P}_l) \subset \mathcal{M}(\mathcal{P}_l)$, where the last inclusion is by Observation 1.3. Applying this mapping for every $l \in L$ extends a solution $(q, y, v) \in \mathcal{M}(\mathbb{PQ})$ to a point (p, y, w). It is straightforward to verify that $w_{lkj} = \sum_{i \in I_l} \lambda_{ik} v_{ilj}$ preserves the commodity balance constraints (4a) and output spec requirements (5). Since $\text{conv}(\mathcal{Q}_l)$ linearly maps to $\text{conv}(\mathcal{P}_l)$ and is thus a tighter relaxation than $\mathcal{M}(\mathcal{P}_l)$, it follows that $\mathcal{M}(\mathbb{PQ})$ is a stronger relaxation than $\mathcal{M}(\mathbb{P})$.

The single pool argument that we adopted here extends to the hybrid formulation where the relaxations corresponding to pools with proportion variables are stronger than the relaxations of these pools in the p-formulation. Hence, the strength of $\mathcal{M}(\mathbb{HYB})$ must be between $\mathcal{M}(\mathbb{P})$ and $\mathcal{M}(\mathbb{PQ})$. □

It is important to note here that there is no relationship between $\mathcal{M}(\mathbb{P})$ and $\mathcal{M}(\mathbb{Q})$.

The convex hull argument in the proof of Proposition 1.3 tells us that applying McCormick envelopes to $\mathbb{PQ}$ along with some additional RLT inequalities, as given in (17), produces the tightest convex relaxation for each pool in a pooling problem. Hence, as far as single pool relaxations are concerned, the pq-relaxation $\mathcal{M}(\mathbb{PQ})$ is the best possible relaxation. This is irrespective of the constraints defining $\mathcal{F}_l$, as long as $\text{conv}(\mathcal{Q}_l)$ in Proposition 1.2 is suitably modified. Thus, the statement of Proposition 1.3 is slightly stronger than the statement of Tawarmalani and Sahinidis [101], Proposition 9.1, who considered only variable bounds on the outflow arcs for standard pooling problems. For generalized pooling problems, our result dominates that of Alfaki and Haugland [7], Proposition 3, who use traditional methods to prove the inclusion of $\mathcal{M}(\mathbb{PQ})$ inside $\mathcal{M}(\mathbb{P})$.

1.5.2.2 Bilinear equality constraints

We close this section by remarking on the commodity balance constraints in the pooling problem. These constraints are not separable across pools and hence were dropped while studying single pool relaxations. We perform a detailed study of the properties of similar bilinear equality constraints and propose new relaxations in Chapter 2. First, we address the issue of polyhedrality of the related feasible sets at each pool. Let $\tilde{\mathcal{P}}_l$ be the set containing constraints that define $\mathcal{P}_l$ and tracking (4a) and incoming flow variables $y_{il}, \forall i \in I \cup L$.

Similarly for $\tilde{\mathcal{Q}}_l$. In the case of generalized pooling problems, the convex hulls of $\tilde{\mathcal{P}}_l$ and $\tilde{\mathcal{Q}}_l$ are unlikely to be polyhedral sets due to the presence of bilinear equalities (4a) and (8), respectively. In the case of standard pooling problems, these complicating bilinear equality constraints are greatly simplified. For $\tilde{\mathcal{Q}}_l$, the commodity balance constraint becomes a defining identity for flows on incoming arcs as $y_{il} = q_{il} \sum_{j \in L \cup J} y_{lj}$ for $i \in I \cup L$ (we dropped the bounds on y_{il} in the definition of $\tilde{\mathcal{Q}}_l$). Then it is easy to show that the convex hull of $\tilde{\mathcal{Q}}_l$, and hence $\tilde{\mathcal{P}}_l$, is a polyhedral set. In fact, in this case, $\text{conv}(\tilde{\mathcal{Q}}_l)$ is given by $\text{conv}(\mathcal{Q}_l)$ and the defining identity $y_{il} = \sum_{j \in L \cup J} v_{ilj}, i \in I_l$.

Second, we discuss some additional relaxation techniques. Only the p-relaxation is considered since all the presented ideas can be extended to the pq-relaxation. For the tracking set defined by (4a) along with bounded flows and bounded specifications, Ruiz and Grossmann [91] developed McCormick envelopes in a different space for this set. The corresponding relaxation may or may not be stronger than $\mathcal{M}(\mathbb{P})$. Now, observe that since $\sum_{j \in L \cup J} p_{lk} y_{lj} = p_{lk} \sum_{j \in L \cup J} y_{lj}$, we can also add a new variable $\tilde{w}_{lk}$ and further impose

$$\sum_{j \in L \cup J} w_{lkj} = \tilde{w}_{lk}, \quad \left(p_{lk}, \textstyle\sum_{j \in L \cup J} y_{lj}, \tilde{w}_{lk}\right) \in \mathcal{M}\left(\{\tilde{w}_{lk} = p_{lk} \textstyle\sum_{j \in L \cup J} y_{lj}\}\right)$$

in the definition of $\mathcal{M}(\mathbb{P})$. This relaxation was also obtained by applying the Reduced RLT (RRLT) procedure in Liberti and Pantelides [67]. Due to the presence of nontrivial upper bound C_l on $\sum_{j \in L \cup J} y_{lj}$, this new relaxation is not necessarily dominated by $\mathcal{M}(\mathbb{P})$. However, it is important to observe that the result of Proposition 1.3 carries through even after tightening the p-relaxation with such a RRLT procedure.

1.5.3 Value function and Lagrangian relaxation

For the standard pooling problem, various Lagrangian relaxations have been proposed over the years. Adhya et al. [3] dualized all constraints except the bounds on p_{lk} and y_{lj} variables in the p-formulation. Almutairi and Elhedhli [9] went one step further by dualizing only the bilinear constraints in the p- and pq-formulations. A more general purpose global optimization algorithm (GOP) based on Lagrangian duals was applied to the Haverly test problems in Visweswaran and Floudas [111].

In this section, we discuss a two-stage value function based approach for solving the pooling problem. This value function is nonconvex and even discontinuous on its domain. Ben-Tal et al. [25] first proposed this value function and used it in a Lagrangian based branch-and-bound algorithm for finding ϵ-optimal solutions; see also Floudas and Aggarwal [40] for a different two-stage algorithm for the p-formulation. Using this value function, we show that the pq-relaxation $\mathcal{M}(\mathbb{PQ})$ is equivalent to a specific Lagrangian dual of the pooling problem. This extends Tawarmalani and Sahinidis [101], Proposition 9.9, to generalized pooling problems and provides a connection between the pq-relaxation and the value function of a pooling problem.

Let η^* denote the optimum value of the pooling problem. Suppose that we use the q-formulation ($\mathbb{Q}$) for solving the problem. Hence,

$$\begin{aligned} \eta^* = \min_{y,p} \quad & \sum_{(i,j)\in\mathcal{A}} c_{ij} y_{ij} \\ \text{s.t.} \quad & y \in \mathcal{F}, \quad (7)-(9). \end{aligned}$$

Note that $y = \mathbf{0}$ always being a feasible flow to the pooling problem implies $\eta^* \leq 0$. Define a value function $\phi\colon \prod_{l\in L} \Delta^{|I_l|} \mapsto \Re$ such that for $q_l \in \Delta^{|I_l|}, \forall l \in L$, the function value $\phi(q)$ is the optimal value of the following linear program

$$\begin{aligned} \phi(q) \quad = \quad \min_{y} \quad & \sum_{(i,j)\in\mathcal{A}} c_{ij} y_{ij} \\ \text{s.t.} \quad & \sum_{j\in L\cup J} y_{lj} \leq C_l, l \in L, \quad 0 \leq y_{ij} \leq u_{ij}, (i,j) \in \mathcal{A} \\ & (2a), (2c), (8), (9) \end{aligned} \tag{18}$$

Since (18) is bounded and $y = \mathbf{0}$ is always feasible to it, $\phi(\cdot)$ is finite valued. The pooling problem can then be equivalently stated as the following global optimization problem

$$\eta^* = \min_q \{\phi(q)\colon q_l \in \Delta^{|I_l|},\ l \in L\}.$$

In the linear program (18) that defines $\phi(q)$, the parameter q appears on both the left and right hand side of the constraints. Hence, not only is $\phi(\cdot)$ nonsmooth but it may also be discontinuous over its domain. We graphically illustrate this function in Example 1.1.

Now consider the pq-relaxation $\mathcal{M}(\mathbb{PQ})$. If $\eta^{\mathbb{PQ}} \leq \eta^*$ is the optimal value of this relaxation, then this lower bound is given by

$$\eta^{\mathbb{PQ}} = \min_q \{\phi_{\mathcal{M}}(q) \colon q_l \in \Delta^{|I_l|}, \forall l \in L\}, \tag{19}$$

where $\phi_{\mathcal{M}}(q) \leq \phi(q)$ is the value function obtained by substituting every term $q_{il}y_{lj}$ in (18) with a new variable v_{ilj} and adding McCormick envelopes (13) for $v_{ilj} = q_{il}y_{lj}$ and the valid inequalities (10). Thus, $\phi_{\mathcal{M}}(q)$ is a value function of a bounded minimization linear program and hence polyhedral.

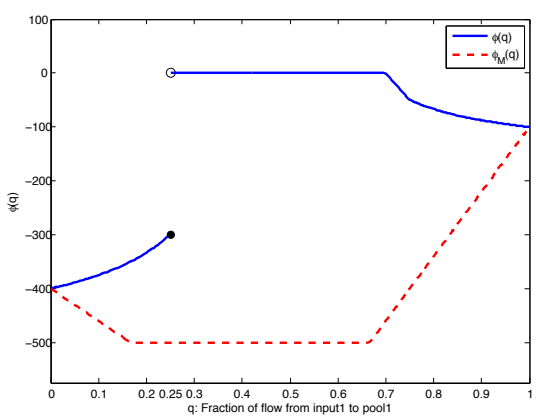
Example 1.1. Consider the Haverly test problem [60]. This is a standard pooling problem with 3 inputs, 1 pool, 2 outputs, and 1 specification. There are three instances of this type [cf. 101]. The solitary pool accepts flows from the first two inputs, whereas the third input is connected directly to the two outputs. Hence $q_1 + q_2 = 1$. We plot $\phi(q_1)$ and $\phi_{\mathcal{M}}(q_1)$ in Figure 3. □

Since $\phi(\cdot)$ is given by a linear program, strong duality of linear programming dictates that $\phi(\cdot)$ is equal to the Lagrangian bound obtained by dualizing (2a), (2c), (8), and (9) with Lagrangian multipliers τ, ρ, Ω, and σ, respectively. Observe that the remaining constraints (2b) and (3) are separable across pools. To simplify our discussion, for every given set of multipliers $(\tau, \rho, \Omega, \sigma)$ and pool $l \in L$, we denote $\psi_l(\tau, \rho, \Omega, \sigma, \cdot)$ to be a affine function in $y_{l\cdot}$, $\xi_l(\Omega, \sigma, \cdot, \cdot)$ to be a bilinear function in q_l and $y_{l\cdot}$, and $\varphi(\tau, \rho, \Omega, \sigma, \cdot)$ to be a affine function in flows on arcs $\{y_{ij}\}$ that did not originate from a pool. The Lagrangian problem becomes

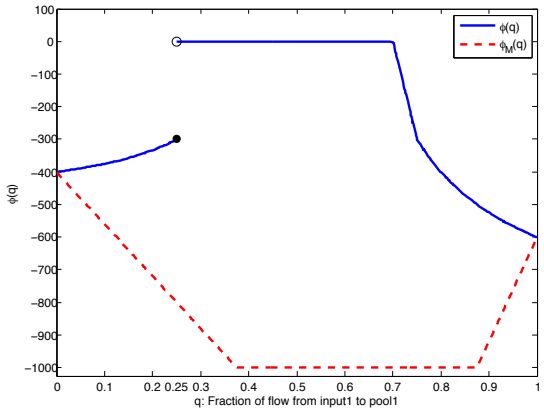
$$\begin{aligned}\phi(q) \quad = \quad \max_{\tau,\rho,\sigma \geq \mathbf{0},\, \Omega} \quad \min_{y} \quad & \varphi(\tau, \rho, \Omega, \sigma, \{y_{ij}\}) \; + \; \sum_{l \in L} \psi_l(\tau, \rho, \Omega, \sigma, y_{l\cdot}) \; + \; \sum_{l \in L} \xi_l(\Omega, \sigma, q_l, y_{l\cdot}) \\ \text{s.t.} \quad & \sum_{j \in L \cup J} y_{lj} \leq C_l, l \in L, \quad 0 \leq y_{ij} \leq u_{ij}, (i,j) \in \mathcal{A}.\end{aligned}$$

Hence the global optimum is

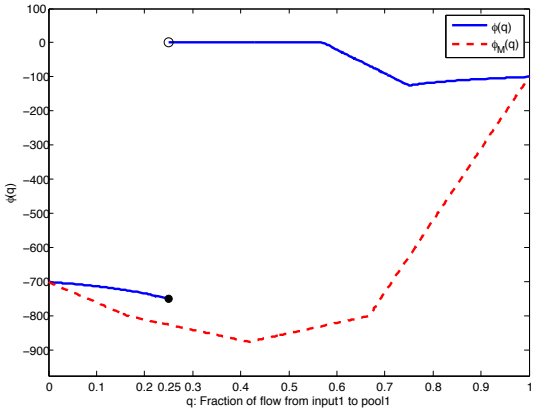
$$\begin{aligned}\eta^* \quad = \quad \min_{\substack{q \colon \\ q_l \in \Delta^{|I_l|}}} \quad \max_{\tau,\rho,\sigma \geq \mathbf{0},\, \Omega} \quad \min_{y} \quad & \varphi(\tau, \rho, \Omega, \sigma, \{y_{ij}\}) \; + \; \sum_{l \in L} \psi_l(\tau, \rho, \Omega, \sigma, y_{l\cdot}) \\ & + \; \sum_{l \in L} \xi_l(\Omega, \sigma, q_l, y_{l\cdot}) \\ \text{s.t.} \quad & \sum_{j \in L \cup J} y_{lj} \leq C_l, l \in L, \quad 0 \leq y_{ij} \leq u_{ij}, (i,j) \in \mathcal{A}\end{aligned}$$



(a) Haverly1



(b) Haverly2



(c) Haverly3 : Discontinuity at optimal solution.

Figure 3: Value functions $\phi(\cdot)$ (solid line) and $\phi_{\mathcal{M}}(\cdot)$ (dotted line) for Haverly instances. The global value function $\phi(\cdot)$ is lower semicontinuous at the point $q = 0.25$ for all three instances. Also, $\eta^{\mathbb{PQ}} < \eta^*$ for all three instances.

By saddle point duality, interchanging the outermost min and max produces a lower bound $\hat{\eta}$ on the optimal value η^*,

$$\begin{aligned}\eta^* \geq \hat{\eta} \quad = \quad \max_{\tau,\rho,\sigma \geq \mathbf{0},\, \Omega} \quad \min_{\substack{q: \\ q_l \in \Delta^{|I_l|}}} \quad \min_{y} \quad & \varphi(\tau,\rho,\Omega,\sigma,\{y_{ij}\}) \;+\; \sum_{l\in L} \psi_l(\tau,\rho,\Omega,\sigma,y_{l\cdot}) \\ & + \sum_{l\in L} \xi_l(\Omega,\sigma,q_l,y_{l\cdot}) \\ \text{s.t.} \quad & \sum_{j\in L\cup J} y_{lj} \leq C_l, l \in L, \quad 0 \leq y_{ij} \leq u_{ij}, (i,j) \in \mathcal{A}\end{aligned}$$

Clearly, $\hat{\eta}$ is a Lagrangian lower bound for the pooling problem. Combining the two inner

minimization problems implies that

$$\begin{aligned}\hat{\eta} \;=\; \max_{\tau,\rho,\sigma\geq\mathbf{0},\,\Omega} \;\; \min_{q,y} \;\; & \varphi(\tau,\rho,\Omega,\sigma,\{y_{ij}\}) \;+\; \sum_{l\in L}\psi_l(\tau,\rho,\Omega,\sigma,y_{l\cdot}) \;+\; \sum_{l\in L}\xi_l(\Omega,\sigma,q_l,y_{l\cdot}) \\ & \text{s.t.} \quad \sum_{j\in L\cup J} y_{lj} \leq C_l, l\in L, \quad 0\leq y_{ij}\leq u_{ij}, (i,j)\in\mathcal{A}, \quad q_l\in\Delta^{|I_l|}, l\in L\end{aligned}$$

Since the inner minimization problem is decomposable across pools,

$$\begin{aligned}\hat{\eta} \;=\; \max_{\tau,\rho,\sigma\geq\mathbf{0},\,\Omega} \;\; \min_{\{y_{ij}\}} \;\; & \varphi(\tau,\rho,\Omega,\sigma,\{y_{i'j'}\}) \;+\; \sum_{l\in L}\; \min_{q_l,y_{l\cdot}} \;\; \psi_l(\tau,\rho,\Omega,\sigma,y_{l\cdot}) \;+\; \xi_l(\Omega,\sigma,q_l,y_{l\cdot}) \\ & \text{s.t.} \quad 0\leq y_{i'j'}\leq u_{i'j'} \qquad\qquad \text{s.t.} \quad y_{l\cdot}\in\mathcal{F}_l, \quad q_l\in\Delta^{|I_l|}\end{aligned}$$

where $\mathcal{F}_l$ is set of feasible outgoing flows from pool l, defined in (14). Since $\xi(\Omega,\sigma,\cdot,\cdot)$ is a bilinear function, the innermost minimization can be solved by replacing each bilinear term $q_{il}y_{lj}$ with a new variable v_{ilj} and enforcing new constraints $v_{ilj} = q_{il}y_{lj}$. With a slight abuse of notation, we denote this transformed function by $\xi(\Omega,\sigma,\cdot)$. Hence

$$\begin{aligned}\hat{\eta} \;=\; \max_{\tau,\rho,\sigma\geq\mathbf{0},\,\Omega} \;\; \min_{\{y_{ij}\}} \;\; & \varphi(\tau,\rho,\Omega,\sigma,\{y_{i'j'}\}) \;+\; \sum_{l\in L}\; \min_{q_l,y_{l\cdot}} \;\; \psi_l(\tau,\rho,\Omega,\sigma,y_{l\cdot}) \;+\; \xi_l(\Omega,\sigma,v_{\cdot l\cdot}) \\ & \text{s.t.} \quad 0\leq y_{i'j'}\leq u_{i'j'} \qquad\qquad \text{s.t.} \quad y_{l\cdot}\in\mathcal{F}_l,\; q_l\in\Delta^{|I_l|},\; v_{ilj}=q_{il}y_{lj}, \forall i,j\end{aligned}$$

The innermost minimization is now exactly the problem of minimizing a linear function in $(q_l, y_{\cdot}, v_{\cdot l\cdot})$ over the set $\mathcal{Q}_l$. The optimum of this problem must lie at an extreme point of $\text{conv}(\mathcal{Q}_l)$, which is described in Proposition 1.2.

$$\begin{aligned}\hat{\eta} \;=\; \max_{\tau,\rho,\sigma\geq\mathbf{0},\,\Omega} \;\; \min_{\{y_{ij}\}} \;\; & \varphi(\tau,\rho,\Omega,\sigma,\{y_{i'j'}\}) \;+\; \sum_{l\in L}\; \min_{q_l,y_{l\cdot}} \;\; \psi_l(\tau,\rho,\Omega,\sigma,y_{l\cdot}) \;+\; \xi_l(\Omega,\sigma,v_{\cdot l\cdot}) \\ & \text{s.t.} \quad 0\leq y_{i'j'}\leq u_{i'j'} \qquad\qquad \text{s.t.} \quad \text{Inequalities from (17)}\end{aligned}$$

Finally, we observe that the above problem is a Lagrangian dual of the pq-relaxation $\mathcal{M}(\mathbb{PQ})$, and hence by strong duality of linear programming, its value must be equal to $\eta^{\mathbb{PQ}}$, i.e. $\hat{\eta} = \eta^{\mathbb{PQ}}$. Thus we have shown the following result.

Proposition 1.4. *Consider the Lagrangian dual of the pooling problem* $(\mathbb{PQ})$ *obtained by dualizing all constraints except the ones in* $\mathcal{F}_l, \Delta^{|I_l|}$*, for all* $l\in L$*, and the variable bounds* $0\leq y_{ij}\leq u_{ij}$ *for every arc* $(i,j)\in\mathcal{A}$*. Then the lower bound provided by this dual is equal to that due to* $\mathcal{M}(\mathbb{PQ})$*.*

1.6 Summary

In this chapter, we have formally introduced the pooling problem and described various optimization formulations of this problem. These formulations were shown to be equivalent and their sizes were compared in terms of number of variables and constraints. A new variant of the pooling problem, with combinatorial constraints, was proposed. Polyhedral relaxations, constructed using envelopes of each bilinear term, were reviewed. Stronger results were presented for comparing these relaxations.

CHAPTER II

BILINEAR SINGLE NODE FLOW

2.1 Introduction

In this chapter, we study the bilinear equality constraints that arise in a pooling problem. In particular, we investigate relaxations of the set

$$\mathcal{P} := \Big\{(x,y) \in \Re^{n+1} \times \Re^{n+m+1} \colon \sum_{i=1}^{n} x_i y_i + \sum_{j=1}^{m} a_j y_{n+j} = x_0 y_0 \tag{20a}$$

$$\sum_{i=1}^{n+m} y_i = y_0 \tag{20b}$$

$$0 \le x_i \le 1, \;\; i = 0, \ldots, n \tag{20c}$$

$$0 \le y_i \le u_i, \;\; i = 0, \ldots, n+m \Big\}. \tag{20d}$$

$\mathcal{P}$ is a nonconvex set in $\Re_+^{n+1} \times \Re_+^{n+m+1}$ defined by a bilinear equality constraint, a linear equality constraint, and finite lower and upper bounds on the variables. The nonconvexity in $\mathcal{P}$ arises from the presence of bilinear terms of the form xy. n and m are integers such that $n \ge 1$ and $m \ge 0$. We make the following assumptions on the data.

Assumption 2.1. 1. $0 < u_i < +\infty$ for all $i = 0, \ldots, n+m$.

2. $u_0 \ge \max\{u_i \colon i = 1, \ldots, n+m\}$, since otherwise (20b) implies that we can replace u_j with u_0 for j such that $u_j > u_0$.

3. $0 \le a_j \le 1$ for all $j = 1, \ldots, m$. This assumption is without loss of generality (w.l.o.g.) since we can scale (20a) with $a_{\max} = \max\{a_j \colon j = 1, \ldots, m\}$ and modify u appropriately.

The interest in studying $\mathcal{P}$ is motivated by network flow problems where total flow on each arc is composed of individual components that must observe mass balance requirement at a intermediate node. Consider Figure 4 that illustrates a single node subsystem in this network.

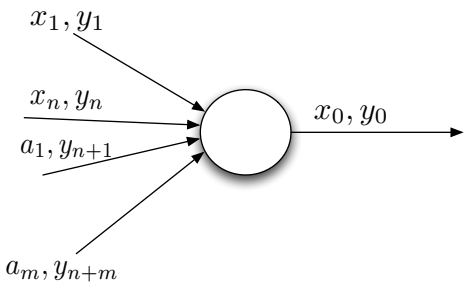


Figure 4: Tracking a single flow component at a node.

There are $n + m$ incoming arcs into this node. For now let us assume that there is a single outgoing arc. Each incident arc i allows a total flow y_i up to its capacity u_i, for $i = 0, \ldots, n + m + 1$. Suppose that the flow on each incoming arc contains one component that we are interested in tracking at this node. For the first n incoming arcs, the relative amount of this component is unknown and hence designated by a variable x_i, for $i = 1, \ldots, n$. For each of the next m arcs, the relative amount of this component is a known nonnegative value $a_i, i = 1, \ldots, m$. Thus, the total flow of this component on arc i is given by $x_i y_i$, for $i = 1, \ldots, n$, and by $a_{i-n} y_i$, for $i = 1 = n + 1, \ldots, n + m$. The node mixes all of the inlet flows to produce a total outflow y_0. The relative amount of this component in the outflow is designated by the variable x_0. Then, equation (20a) imposes the condition that the total quantity of this component must be conserved at this node. Similarly, (20b) maintains total flow balance at this node. Variable bounds for x and y are given by (20c) and (20d), respectively. Note that the upper bound on x_i is 1, for $i = 1, \ldots, n$. This assumption is w.l.o.g. up to scaling of x and a. The important assumption here is that all the x variables have the same upper bound. The upper bound $x_0 \leq 1$ is implied because $x_i \leq 1, i = 1, \ldots, n$, and $a_j \leq 1, j = 1, \ldots, m$, by assumption, and (20a) along with (20b) can be interpreted as expressing x_0 being a convex combination of $\{x_1, \ldots, x_0, a_1, \ldots, a_m\}$. Finally, we comment that for nodes with multiple outgoing arcs, since each outflow carries the same relative amount x_0 of this component, we can aggregate the outgoing flows in to a single arc, thereby obtaining a relaxation $\mathcal{P}$.

The set $\mathcal{P}$ often arises in chemical processing networks [85], particularly in the pooling problem of Chapter 1 (cf. (4a) and (8)). In these applications, the x variables may typically

signify concentrations of chemical compounds such as sulphur, carbon, or physical properties such as density, octane number, etc. Other application areas include wastewater systems [64], distillation sequences [91], and heat exchanger networks [41].

In this chapter, we are interested in deriving strong relaxations of $\mathcal{P}$. For this purpose, we study different ways of obtaining linear inequalities that are valid for the convex hull of $\mathcal{P}$, henceforth denoted by $\text{conv}(\mathcal{P})$. Since $\mathcal{P}$ arises at each intermediate node of a network flow problem, it constitutes an important substructure of the overall problem. For mixed integer linear problems, valid inequalities derived from well structured relaxations have proven a useful tool for solving these problems, see for e.g. [31, 34, 52, 116]. Since $\mathcal{P}$ is a nonconvex set, any optimization problem involving $\mathcal{P}$ must be solved by a global optimization solver. Popular global solvers, such as BARON [93, 104] and Couenne [23], use the branch-and-cut algorithm that relies on building tight polyhedral relaxations of the problem at each node of the search tree. Although more general convex relaxations have also been studied [19, 27, 94], polyhedral relaxations remain a common choice due to faster solution times and easier warm-starting and cut management of the associated linear programs. The computational benefits of using linearization strategies for solving different classes of global optimization problems can be found in [1, 20, 104, 109].

Although well known relaxation methods exist for bilinear constraints, these methods have two potential drawbacks : first, they use convex and/or concave envelopes of a bilinear function to relax the bilinear constraint. However, since for any function $f(\chi)$, in general the convex hull of $\{\chi\colon f(\chi) \leq 0\}$ is not equal to $\{\chi\colon \text{cvx}\, f(\chi) \leq 0\}$, where $\text{cvx}\, f(\cdot)$ is the convex envelope of $f(\cdot)$ taken over its domain, such a envelope-based relaxation may prove to be weak. Secondly, relaxing a bilinear function $f(\cdot)$ with its convex and/or concave envelopes involves adding linear number of extra variables and possibly solving a linear program to iteratively separate facets of its envelope. Our focus is on obtaining closed form expressions for valid inequalities to $\mathcal{P}$ in the original (x, y)-space. We derive these inequalities using lifting. Lifting is a well known technique from MILP that generates a strong inequality for a set by suitably transforming a linear inequality that is valid for a restriction of this given set. Initial work in lifting can be traced back to Wolsey [114]. Wolsey [115] and Gu et al.

[53] proved that if a certain function is superadditive, then all the fixed variables can be lifted in one step to make the inequality globally valid. See Louveaux and Wolsey [68] for a review on lifting and Richard et al. [88] for lifting of continuous variables. Atamtürk and Narayanan [13] derived lifted conic inequalities for conic MILPs. Recently, Richard and Tawarmalani [87] provided a generalization of the lifting procedure to nonlinear problems. Their results were applied to study the convex hulls of mixed integer bilinear knapsacks [87] and mixed integer bilinear covering sets [33].

This chapter is structured as follows. In §2.2 we discuss basic properties of $\mathcal{P}$. In particular, we demonstrate that its convex hull may not be a polyhedral set. Then, we briefly turn our attention to a variant of $\mathcal{P}$, denoted as $\mathcal{Q}$, obtained by introducing multiple outgoing arcs and relaxing flow balance (20b). We present two types of sufficient conditions under each of which the convex hull of $\mathcal{Q}$ is polyhedral. Then, we explicitly enumerate all the nontrivial extreme points of $\mathcal{P}$. These extreme points help define restrictions of $\mathcal{P}$ that will be useful later in our analysis. They also provide sufficient conditions under which the convex hull of $\mathcal{P}$ is polyhedral. Extended polyhedral relaxations of $\mathcal{P}$ using McCormick envelopes [70] are addressed in §2.3. For the set $\mathcal{Q}$, we prove that if the coefficients a are all equal to one, then under the sufficient conditions for $\text{conv}(\mathcal{Q})$ to be polyhedral, these McCormick envelopes provide the strongest possible relaxations. §2.4 studies restrictions of $\mathcal{P}$ in (x_0, y_j)-space for some j and obtained by fixing the remaining variables at values taken by them at extreme points. We use these restrictions to provide a disjunctive representation, that is also conic quadratic representable, for the convex hull of $\mathcal{P}$. We present a countable family of polyhedral relaxations of $\mathcal{P}$ and show that each member of this family is stronger (under inclusion) than the McCormick relaxations. In §2.5, we construct valid linear inequalities to the convex hull of $\mathcal{P}$ via lifting. Since our set $\mathcal{P}$ is neither mixed integer linear nor does it have a bilinear packing/covering structure, we extend the lifting theory to our case by proving analogous counterparts of the classical results for sequence independent lifting. The algebraic proofs we provide are inspired by the classical approach and suitably modified to meet our needs. We then use these results to derive two exponential families of valid inequalities in the (x, y)-space.

We adopt the following notation in the rest of this chapter : conv$(\cdot)$ denotes the convex hull of a set and ext$(\cdot)$ is its set of extreme points. $\mathbf{e}$ is a vector of ones, $\mathbf{0}$ is a vector of zeros, and $\mathbf{e}_i$ is the i^{th} unit vector. $\Re$ is the set of reals and $\mathbb{Z}$ the set of integers. $\text{Proj}_x \cdot$ is the projection operator onto the x-space and θ^+ denotes $\max\{0, \theta\}$ for $\theta \in \Re$.

2.2 Basic properties of $\mathcal{P}$

In this section, we study basic properties of $\mathcal{P}$. In particular, we are interested in characterizing the convex hull of $\mathcal{P}$ using its extreme points. Since the bilinear function is continuous, $\mathcal{P}$ is a closed set. Hence $\mathcal{P}$ is compact since it is also bounded by assumption. This implies conv$(\mathcal{P})$ can be written as a convex hull of its extreme points.

Observation 2.1. conv$(\mathcal{P})$ *is a compact set.*

Although conv$(\mathcal{P})$ is compact, it is not always polyhedral. We illustrate this using a simple example.

Observation 2.2. conv$(\mathcal{P})$ *may not be a polyhedral set.*

Proof. Consider the following example with $n = 2$ and $m = 1$:

Example 2.1.

$$\mathcal{P} = \Big\{(x, y) \in \Re^3 \times \Re^3 \colon x_1 y_1 + x_2 y_2 + y_3 = x_0 y_0,\ y_1 + y_2 + y_3 = y_0$$
$$0 \le x_0, x_1, x_2 \le 1, 0 \le y_1 \le 3, 0 \le y_2 \le 2, 0 \le y_3 \le 1, 0 \le y_0 \le 6\Big\}.$$

Fix $x_1 = 0, x_2 = 1, y_1 = 3$, and $y_3 = 1$, i.e. at one of their respective extremal values. This gives a restriction of $\mathcal{P}$ in $\Re^2$ after projecting out y_0. Let this restriction be $\mathcal{F} = \{(x_0, y_2) \colon y_2 = 4x_0/(1 - x_0), 0 \le x_0 \le 1, 0 \le y_2 \le 2\}$. Then $x_0 \le 1/3$ is valid to $\mathcal{F}$ and it is easy to verify that $\mathcal{F}$ is given by a convex curve such that every $(x_0, y_2) \in \mathcal{F}$ is an extreme point of conv$(\mathcal{F})$. Since the remaining variables were fixed at their bounds, it follows that every $(x_0, y_2) \in \mathcal{F}$ corresponds to an extreme point of conv$(\mathcal{P})$. Thus conv$(\mathcal{P})$ has infinitely many extreme points. □

Going forward in this section, we would like to study sufficient conditions under which the convex hull of $\mathcal{P}$ is a polyhedral set. Towards this end, we first study the effect of relaxing the flow conservation constraint (20b) in $\mathcal{P}$.

2.2.1 Relaxing flow conservation

We address a case in which $\text{conv}(\mathcal{P})$ is polyhedral. This case relies on relaxing the flow conservation identity (20b) in $\mathcal{P}$ and that the upper bounds on all the y variables are equal. We prove this result for a more general set using the following simple lemma.

Lemma 2.1. *Consider the set $\mathcal{Q}$ defined as follows,*

$$\begin{aligned}\mathcal{Q} := \Big\{(x,y)\colon \sum_{i=1}^{n_1} c_i x_i y_i - \sum_{i=n_1+1}^{n_1+n_2} c_i x_i y_i + \sum_{j=1}^{m_1} a_j y_{n_1+n_2+j} - \sum_{j=m_1+1}^{m_1+m_2} a_j y_{n_1+n_2+j} = b \\ \tilde{l}_i \le x_i \le \tilde{u}_i,\ \ i = 1,\dots,n_1+n_2 \\ l_i \le y_i \le u_i,\ \ i = 1,\dots,n_1+n_2+m_1+m_2\Big\},\end{aligned} \tag{21}$$

where $a, c > \mathbf{0}$, $\tilde{l} < \tilde{u}$, and $l < u$. If $x_j \in (\tilde{l}_j, \tilde{u}_j)$ and $x_k \in (\tilde{l}_k, \tilde{u}_k)$ (or $y_k \in (l_k, u_k)$) for indices $j \neq k$, then (x,y) cannot be an extreme point of $\text{conv}(\mathcal{Q})$.

Proof. Consider an extreme point of $\text{conv}(\mathcal{Q})$, denoted by (x,y). Since (x,y) is an extreme point, it must be in $\mathcal{Q}$. We will show that it is not possible to have $x_j \in (\tilde{l}_j, \tilde{u}_j)$ and $x_k \in (\tilde{l}_k, \tilde{u}_k)$ for $j \neq k$. A similar argument carries through when x_k is replaced by y_k. Assume that $j, k \le n_1$. The other case with $j \le n_1, n_1 < k \le n_1 + n_2$ is analogous.

Since our chosen point satisfies the bilinear equality constraint, we can rewrite

$$\begin{aligned}c_j x_j y_j + c_k x_k y_k &= b - \sum_{\substack{i=1 \\ i\neq j,k}}^{n_1} c_i x_i y_i + \sum_{i=n_1+1}^{n_1+n_2} c_i x_i y_i - \sum_{j=1}^{m_1} a_j y_{n_1+n_2+j} + \sum_{j=m_1+1}^{m_1+m_2} a_j y_{n_1+n_2+j} \\ &= b - \sigma,\end{aligned}$$

where σ is used for ease of notation.

If either $y_j = 0$ or $y_k = 0$, then we can write x_j or x_k as convex combination of $x_j \pm \epsilon$ or $x_k \pm \epsilon$, respectively. Now suppose that both y_j and y_k are nonzero. Construct two new points $(\bar{x}, \bar{y})$ and $(\hat{x}, \hat{y})$ as follows,

$$\begin{aligned}&\bar{x}_k = x_k + \epsilon && \hat{x}_k = x_k - \epsilon \\ &\bar{x}_j = x_j - \tfrac{\epsilon c_k y_k}{c_j y_j} && \hat{x}_j = x_j + \tfrac{\epsilon c_k y_k}{c_j y_j} \\ &\bar{x}_i = x_i \qquad i \neq j,k && \hat{x}_i = x_i \qquad i \neq j,k \\ &\bar{y} = y && \hat{y} = y.\end{aligned}$$

Since y_j is nonzero by assumption and $c_j > 0$, these two points are well defined. For a sufficiently small $\epsilon > 0$, the two points $(\bar{x}, \bar{y})$ and $(\hat{x}, \hat{y})$ are guaranteed to lie within their respective bounds. They also satisfy (20a) because

$$\begin{aligned} c_j\bar{x}_j\bar{y}_j + c_k\bar{x}_k\bar{y}_k &= c_j\hat{x}_j\hat{y}_j + c_k\hat{x}_k\hat{y}_k \\ &= c_jx_jy_j + c_kx_ky_k \\ &= b - \sigma \end{aligned}$$

Hence, the two new points belong to $\mathcal{Q}$ and we can express $(x, y) = \frac{1}{2}(\bar{x}, \bar{y}) + \frac{1}{2}(\hat{x}, \hat{y})$. Thus, we have proved that (x, y) cannot be an extreme point of conv $\mathcal{Q}$.

Similarly, we can address the remaining cases : 1) $x_j \in (\tilde{l}_j, \tilde{u}_j), y_k \in (l_k, u_k)$ or 2) $y_j \in (l_j, u_j)$ and $y_k \in (l_k, u_k)$. □

Lemma 2.1 helps us eliminate from the candidate list of extreme points of conv$(\mathcal{Q})$ those points where variables from two distinct terms take non-extreme values. This does not guarantee though that there are finitely many extreme points of conv$(\mathcal{Q})$. A sufficient condition for the polyhedrality of conv$(\mathcal{Q})$ is provided in §2.7. We next consider a special case of $\mathcal{Q}$ that relates to $\mathcal{P}$.

Proposition 2.1. *Consider the set $\mathcal{Q}$ in* (21) *and assume the following*

1. $\tilde{l} = l = \mathbf{0}$, *and* $\tilde{u} = \mathbf{e}$, $u = U\mathbf{e}$ *for some* $U \in \mathbb{Z}_{++}$,
2. $c = \mathbf{e}$, *and* $a \in \mathbb{Z}_{++}^{m_1+m_2}$,
3. $b \in \mathbb{Z}$ *such that* $b \equiv 0 \mod U$.

Then conv$(\mathcal{Q})$ *is a polyhedral set.*

In particular, if $(x, y) \in \text{ext}(\text{conv}(\mathcal{Q}))$, *then* $x_i \in \{0, 1\}, y_i \in \{0, U\}$, *for all* $i = 1, \ldots, n_1 + n_2$, *and* $a_j y_{n_1+n_2+j} \in \mathbb{Z}$, *for all* $j = 1, \ldots, m_1 + m_2$.

Proof. The assumptions that $u_j = U, \forall j$, and U divides b allow scaling the y variables to make all their upper bounds equal to 1 and obtain a set similar to $\mathcal{Q}$ but with a different integral right hand side $b' = b/U$. Hence, it suffices to prove the result for $U = 1$. So

let us assume that $U = 1$. Lemma 2.1 implies that no two variables from different terms can take values within $(0, 1)$. Now suppose that $x_1, y_1 \in (0, 1)$. Then by Lemma 2.1 it must be that the remaining variables are fixed at one of their bounds. Let $x_1 y_1 = b + \sigma$, where $\sigma \in \mathbb{Z}$ due to integral bounds on variables, $c_i = 1, \forall i$, and $a_j \in \mathbb{Z}_{++}, \forall j$. Hence $b + \sigma \in \mathbb{Z}_+$, since $b \in \mathbb{Z}$. It follows that neither x_1 nor y_1 can take fractional values. Similarly, $a_1 y_{n_1+n_2+1}$ cannot take a fractional value due to the integrality of its coefficient a_1. Hence, at any $(x, y) \in \text{ext}(\text{conv}(\mathcal{Q}))$, there are only finitely many possible values for each variable, implying that $\text{conv}(\mathcal{Q})$ is polyhedral. □

Let us revert back to the set $\mathcal{P}$. Suppose that we relax the flow balance constraints (20b). If we further assume that all incoming arcs and the sole outgoing arc carry the same integral upper bound on flows, then $\mathcal{P}$ becomes a specific case of $\mathcal{Q}$ with $n_2 = 1$,$m_2 = 0$, and $c = \mathbf{e}$. Hence the next result follows immediately from Proposition 2.1.

Corollary 2.1. *If total flow balance* (20b) *is relaxed and $u_i = U$, for all $i = 0, \ldots, n + m$, and some positive integer U, then* $\text{conv}(\mathcal{P})$ *is a polyhedral set.*

2.2.2 Extreme points of $\mathcal{P}$

Henceforth, we assume that y_0 is substituted out in the definition of $\mathcal{P}$ using (20b). Thus, the set $\mathcal{P}$ is expressed as

$$\mathcal{P} = \Big\{(x, y) \in \Re^{n+1} \times \Re^{n+m} \colon \sum_{i=1}^{n} x_i y_i + \sum_{j=1}^{m} a_j y_{n+j} = x_0 \sum_{i=1}^{n+m} y_i \tag{22a}$$

$$\sum_{i=1}^{n+m} y_i \leq u_0 \tag{22b}$$

$$0 \leq x_i \leq 1, \;\; i = 0, \ldots, n \tag{22c}$$

$$0 \leq y_i \leq u_i, \;\; i = 1, \ldots, n + m\Big\}. \tag{22d}$$

Definition 2.1. $(x, y) \in \mathcal{P}$ is said to be a *trivial* extreme point of $\text{conv}(\mathcal{P})$ if $y_j = 0$ for all $j = 1, \ldots, n + m$.

Observation 2.3. *Let (x, y) be a trivial extreme point of* $\text{conv}(\mathcal{P})$*. Then it must be that $x_i \in \{0, 1\}$, for all $i = 0, \ldots, n$, and $y_j = 0$, for all $j = 1, \ldots, n + m$.*

We now characterize all the nontrivial extreme points of $\text{conv}(\mathcal{P})$.

Theorem 2.1. *Any (x, y) is a nontrivial extreme point of $\text{conv}(\mathcal{P})$ only if the following three conditions are satisfied:*

1. *$x_i \in \{0, 1\}$, for all $i = 1, \ldots, n$,*
2. *For some $j \in \{1, \ldots, n+m\}$ and $C \subseteq \{1, \ldots, n+m\} \setminus j$ such that $\sum_{i \in C} u_i \leq u_0$, we have*
   (a) *$y_j \in [0, u_j]$ such that $y_j + \sum_{i \in C} u_i \leq u_0$, and*
   (b) *$y_i = u_i$, for all $i \in C$, $y_i = 0$, for all $i \notin C \cup j$,*
3. *The value of x_0 is given by*

$$x_0 \;=\; \frac{\sum_{i=1}^{n} x_i y_i + \sum_{j=1}^{m} a_j y_{n+j}}{\sum_{i=1}^{n+m} y_i} \;\in [0, 1].$$

Proof. Since $\mathcal{P}$ is compact, we know that $\text{ext}\,\text{conv}(\mathcal{P}) \subseteq \mathcal{P}$. Now consider a extreme point $(x, y) \in \text{ext}\,\text{conv}(\mathcal{P})$. First note that since $\sum_i y_i > 0$, the bilinear equality (20a) along with (20b) implies

$$x_0 \in \text{conv}\{x_1, \ldots, x_0, a_1, \ldots, a_m\}. \qquad (\star)$$

A similar argument as in Lemma 2.1 implies that x_i, x_j cannot be fractional for any two indices i, j. Now suppose that $0 < x_i < 1$ and $x_j \in \{0, 1\}, \forall j \neq i$. If $y_i = 0$, then (x, y) can be written as a convex combination of two new feasible points obtained from (x, y) by setting x_i to either 0 or 1. Now suppose $y_i > 0$. Since $x_i, y_i > 0$ and $x_i < 1$, it follows that $0 < x_0 < 1$. Define $\Delta := \sum_{j=1}^{n+m} y_j > 0$ and δ such that $x_i y_i + \delta = \Delta x_0$. Note that $\delta \geq 0$ due to $a, x, y \geq \mathbf{0}$. Consider two new points constructed from (x, y) by replacing x_i with $x_i \pm \epsilon$ and x_0 with $x_0 \pm \epsilon y_i/\Delta$, for some sufficiently small $\epsilon > 0$. In particular, let $0 < \epsilon \leq \min\{x_i, 1 - x_i, \Delta(1 - x_0)/y_i\}$. Note that since $0 < x_0 < 1$, there exists such a positive ϵ. By construction, these two new points satisfy (20a). It remains to check that the bounds on x_0 are not violated. Since $x_i y_i + \delta - \Delta x_0 = 0$ and $\delta \geq 0$, we get that $x_i y_i - \Delta x_0 \leq 0$. This implies $\epsilon y_i \leq \Delta x_0$ because ϵ is sufficiently small such that $\epsilon \leq x_i$. Equivalently, we have shown that $x_0 - \epsilon y_i/\Delta \geq 0$. Similarly, using $\epsilon \leq \Delta(1 - x_0)/y_i$, we get

that $x_0 + \epsilon y_i/\Delta \leq 1$. Thus, the two new points belong to $\mathcal{P}$. Since we can write (x, y) as a convex combination of these two points, it follows that x_i cannot be fractional within $[0, 1]$ at an extreme point of $\text{conv}(\mathcal{P})$.

Now suppose that $y_i \in (0, u_i)$ and $y_j \in (0, u_j)$ for two distinct indices $i, j \in \{1, \ldots, n\}$. If $x_i = x_j$, then we can write (x, y) as a convex combination of two new feasible points obtained by adjusting y_i and y_j as $y_i \mp \epsilon$ and $y_j \pm \epsilon$. Now suppose that $x_i = 1, x_j = 0$. Consider two new points constructed with $y_i \mp \epsilon, y_2 \pm \epsilon$, and $x_0 \mp \epsilon/\Delta$. By construction, these two points are feasible to (22a). $x_0 - \epsilon/\Delta \geq 0$ since $\Delta x_0 \geq y_i \geq \epsilon$. Also, since $x_j = 0$ and $y_j > 0$, $(\star)$ implies that $x_0 < 1$. Thus $x_0 + \epsilon/\Delta \leq 1$ is true for sufficiently small positive ϵ. Thus the two new points belong to $\mathcal{P}$ implying that y_i and y_j cannot both be within bounds at an extreme point of $\text{conv}(\mathcal{P})$.

Next, we resolve the case that $y_i \in (0, u_i)$ and $y_j \in (0, u_j)$ for $i \in \{1, \ldots, n\}$ and $j \in \{n+1, \ldots, n+m\}$. The arguments are almost similar to the previous case. Construct two points with $y_i \mp \epsilon$ and $y_j \pm \epsilon$. If $a_j = x_i$, then these points are in $\mathcal{P}$. Else if $a_j > x_i = 0$ (recall that we assumed $a_j \in [0, 1]$ for all j), then we also set $x_0 \pm a_j\epsilon/\Delta$. Now, $x_0 - a_j\epsilon/\Delta \geq 0$ because $\Delta x_0 \geq a_j y_j \geq a_j\epsilon$ and $x_0 + a_j\epsilon/\Delta \leq 1$ for small positive ϵ since $x_0 < 1$ from $x_i = 0, y_i > 0$, and $(\star)$. Else, $a_j < x_i = 1$. Then we set $x_0 \pm (a_j - 1)\epsilon/\Delta$. Again, $0 < x_0 < 1$. Hence, for $0 < \epsilon \leq \Delta \min\{x_0, 1 - x_0\}/(1 - a_j)$, we get that the bounds on x_0 are satisfied.

The case where both $i, j \in \{n+1, \ldots, n+m\}$ is identical to the previous case with x_i and its corresponding value in $\{0, 1\}$ replaced by a_i.

□

The extreme point characterization of Theorem 2.1 helps us to strengthen the statement of Corollary 2.1. In particular, it allows us to avoid relaxing total flow balance constraint.

Corollary 2.2. *Consider the set $\mathcal{P}$. Let U be some positive integer and suppose that $u_i = U$, for all $i = 0, \ldots, n+m$. Then* $\text{conv}(\mathcal{P})$ *is a polyhedral set.*

Proof. Consider an extreme point (x, y) and some variable y_j. From Theorem 2.1, we know that $y_k \in \{0, U\}$, for all $k \neq j$, such that $y_j + \sum_{k \neq j} y_k \leq U$. Clearly, then it must be that at most one other variable y_i can take the value U. If $y_i = U$ for some $i \neq j$, then

$y_j = 0$. Otherwise, $y_k = 0$ for all $k \neq j$. Then, since (x, y) satisfies (22a), we must have $x_j y_j = x_0 y_j$. Theorem 2.1 implies $x_j \in \{0, 1\}$. Hence, either $x_0 y_j = 0$ or $(1 - x_0) y_j = 0$. It is straightforward to verify that $y_j \in \{0, U\}$ at extreme points in both these cases. Thus, $\text{conv}(\mathcal{P})$ can have only finite number of extreme points. □

We now address a variant of $\mathcal{P}$ obtained by restricting flows to take only integer values within their bounds. Define $\mathcal{P}^{\mathcal{I}} := \mathcal{P} \cap (\Re^{n+1} \times \mathbb{Z}_+^{n+m})$.

Proposition 2.2. *Consider $\mathcal{P}^{\mathcal{I}}$ and assume w.l.o.g. that $u_i \in \mathbb{Z}_{++}$ for all i. Then, $\text{conv}(\mathcal{P}^{\mathcal{I}})$ is polyhedral and $(x, y) \in \text{ext conv}(\mathcal{P}^{\mathcal{I}})$ only if*

1. *$x_i \in \{0, 1\}$, for all $i = 1, \ldots, n$,*
2. *For some $j \in \{1, \ldots, n + m\}$ and $C \subseteq \{1, \ldots, n + m\} \setminus j$ such that $\sum_{i \in C} u_i \leq u_0$, we have*
   (a) *$y_j \in [0, u_j] \cap \mathbb{Z}_+$ such that $y_j + \sum_{i \in C} u_i \leq u_0$, and*
   (b) *$y_i = u_i$, for all $i \in C$, $y_i = 0$, for all $i \notin C \cup j$,*
3. *The value of x_0 is given by*

$$x_0 \;=\; \frac{\sum_{i=1}^{n} x_i y_i + \sum_{j=1}^{m} a_j y_{n+j}}{\sum_{i=1}^{n+m} y_i} \;\in [0, 1].$$

Proof. Clearly, the convex hull of $\mathcal{P}^{\mathcal{I}}$ is a polytope since the y variables are bounded integers and hence $\mathcal{P}^{\mathcal{I}}$ can be written as a disjunction of a finite number of polytopes. Note that the proposed necessary conditions for the extreme points are the same as those in Theorem 2.1 with the added restriction that $y_j \in \mathbb{Z}_+$. Then, to show that all the extreme points can be obtained from Theorem 2.1, it suffices to verify that the choice $\epsilon = 1$ works in all cases addressed in the proof of Theorem 2.1.

We begin by observing that $x_i \in \{0, 1\}$ for all $i = 1, \ldots, n$, since in the first part of the proof of Theorem 2.1, we did not change values of the y variables. Now suppose that $y_i \in (0, u_i)$ and $y_j \in (0, u_j)$ for two distinct indices $i, j \in \{1, \ldots, n\}$. Hence it must be that $y_i, y_j \geq 1$ since $y_i, y_j \in \mathbb{Z}_+$. We have to check the case $x_i = 1, x_j = 0$. Let $\Delta = \sum_{k=1}^{n+m} y_k$ as

before. Also let $\delta = \sum_{\substack{k=1 \\ k \neq i}}^{n} x_k y_k$. Thus,

$$\Delta \;\; = \sum_{k=n+1}^{n+m} y_k + y_i + y_j + \sum_{\substack{k=1 \\ k \neq i,j}}^{n} y_k \;\; \geq \sum_{k=n+1}^{n+m} y_k + y_i + 1 + \delta, \tag{23}$$

since $y_j \geq 1$ and $\sum_{\substack{k=1 \\ k \neq i}}^{n} y_k \geq \delta$ due to $x_k \leq 1$. Equation (22a) and $x_i = 1$ implies $\Delta x_0 \geq y_i \geq 1$ and hence $\epsilon = 1$ is valid. Now we must also show that $\Delta - \Delta x_0 \geq 1$. Towards that end, note that (20a) is

$$\sum_{k=n+1}^{n+m} a_k y_k + y_i + \delta = \Delta x_0. \tag{24}$$

Equation (23) implies

$$\Delta - [y_i + \delta + 1] \;\; \geq \sum_{k=n+1}^{n+m} y_k \;\; \geq \sum_{k=n+1}^{n+m} a_k y_k,$$

where the second inequality is due to $a_k \leq 1$ for all k. After rearranging terms we get that $\Delta - \Delta x_0 \geq 1$.

Now assume that $y_i \in (0, u_i)$ and $y_j \in (0, u_j)$ for $i \in \{1, \ldots, n\}$ and $j \in \{n+1, \ldots, n+m\}$. We only have to check $a_j \neq x_i$. Suppose $a_j > x_i = 0$. Equation (24) implies $\Delta x_0 \geq a_j y_j$ and since $y_j \geq 1$ we have $\Delta x_0 \geq a_j$. To check $\Delta - \Delta x_0 \geq a_j$, we first note that $\Delta \geq \sum_{\substack{k=n+1 \\ k \neq j}}^{n+m} y_k + y_j + 1 + \delta$. Hence $\Delta - (y_j + 1 + \delta) \geq \sum_{\substack{k=n+1 \\ k \neq j}}^{n+m} a_k y_k$ since $a_k \leq 1$. Also, $a_j y_j + a_j \leq y_j + 1$. Thus after rearranging we get the required inequality $\Delta - \Delta x_0 \geq a_j$. For the final case, $a_j < x_i = 1$. First, $\Delta x_0 \geq 1 \geq 1 - a_j$. Second, to show $\Delta - \Delta x_0 \geq 1 - a_j$, the steps of the proof are similar to the previous case ($a_j > x_i = 0$) with a_j replaced by $1 - a_j$. □

Thus we have characterized all the extreme points of $\mathcal{P}$ and $\mathcal{P} \cap (\Re^{n+1} \times \mathbb{Z}_+^{n+m})$. We have shown that these extreme points are obtained by fixing all the x_i variables, for $i = 1, \ldots, n$, to 0 or 1 and all but one of the y variables to either of their respective bounds such that total flow balance (22b) is satisfied. Recall that in the definition of $\mathcal{P}$, all the x and y variables are allowed to take continuous values in their domain. This extreme point characterization enforces a combinatorial aspect to $\mathcal{P}$ that we shall later exploit in §2.4 and §2.5 to build tight relaxations for the convex hull of $\mathcal{P}$.

We close this section by showing that the convex hull of $\mathcal{P}$ has dimension $2n + m + 1$.

Proposition 2.3. $\text{conv}(\mathcal{P})$ *is full-dimensional.*

Proof. Construct the following $2n + m + 2$ points where each point is represented as $\Big(\{x_i\}_{i=1}^n, x_0, \{y_i\}_{i=1}^n, \{y_{n+j}\}_{j=1}^m\Big)$.

1. $(\mathbf{0}, 0, \mathbf{0}, \mathbf{0})$ and $(\mathbf{e}_1, 0, \mathbf{0}, \mathbf{0})$,
2. $(\mathbf{e}_i, 1, u_i\mathbf{e}_i, \mathbf{0})$ and $(\mathbf{e} - \mathbf{e}_i, 0, u_i\mathbf{e}_i, \mathbf{0})$ for $i = 1, \ldots, n$,
3. $(\mathbf{0}, a_j, \mathbf{0}, u_{n+j}\mathbf{e}_j)$ for $j = 1, \ldots, m$.

It is easy to verify that each of the above points belongs to $\mathcal{P}$ and hence $\text{conv}(\mathcal{P})$. Since $u > \mathbf{0}$ by Assumption 2.1, there is only one zero vector in this collection of points. Also, the $2n+m+1$ nonzero points are linearly independent. Hence, we have $2n+m+2$ affinely independent points that belong to $\text{conv}(\mathcal{P})$, thus completing the proof. □

2.3 Standard polyhedral relaxations

For the set $\mathcal{P}$, we can introduce new variables $w_i = x_i y_i$ for $i = 1, \ldots, n$ and $w_0 = x_0 \sum_{j=1}^{n+m} y_j$ and use the McCormick envelopes from (13) to obtain a higher-dimensional polyhedral relaxation $\mathcal{M}(\mathcal{P})$.

$$\begin{aligned}\mathcal{M}(\mathcal{P}) := \Big\{(x, y, \{w_i\}_{i=1}^n, w_0)\colon\ & \sum_{i=1}^{n} w_i + \sum_{j=1}^{m} a_j y_{n+j} = w_0,\ \ \sum_{j=1}^{n+m} y_j \le u_0 \\ & \max\{0, u_i x_i + y_i - u_i\} \le w_i \le \min\{u_i x_i, y_i\},\ \ i = 1, \ldots, n \\ & \max\{0, u_0 x_0 + \sum_{j=1}^{n+m} y_j - u_0\} \le w_0 \le \min\{u_0 x_0, \sum_{j=1}^{n+m} y_j\}\ \Big\}.\end{aligned} \tag{25}$$

Alternatively, we may disaggregate the sum $x_0 \sum_{j=1}^{n+m} y_j$ and introduce new variables $w_{0j} = x_0 y_j$ for all $j = 1, \ldots, n + m$. This produces a different higher-dimensional polyhedral

relaxation $\mathcal{SM}(\mathcal{P})$.

$$\mathcal{SM}(\mathcal{P}) := \Big\{ (x, y, \{w_i\}_{i=1}^n, \{w_{0j}\}_{j=1}^{n+m}) \colon \sum_{i=1}^{n} w_i + \sum_{j=1}^{m} a_j y_{n+j} = \sum_{j=1}^{n+m} w_{0j}, \ \sum_{j=1}^{n+m} y_j \le u_0$$
$$\max\{0, u_i x_i + y_i - u_i\} \le w_i \le \min\{u_i x_i, y_i\}, \ \ i = 1, \dots, n$$
$$\max\{0, u_j x_0 + y_j - u_j\} \le w_{0j} \le \min\{u_j x_0, y_j\}, \ \ j = 1, \dots, n+m \Big\}. \tag{26}$$

$\mathcal{M}(\mathcal{P})$ introduces $n+1$ extra variables whereas $\mathcal{SM}(\mathcal{P})$ introduces $2n+m$ extra variables. Note that we created the two McCormick relaxations $\mathcal{M}(\mathcal{P})$ and $\mathcal{SM}(\mathcal{P})$ using envelopes of a single bilinear term $x_i y_i$. One may expect to derive a stronger relaxation using envelopes of the bilinear function $f(x, y) = \sum_{i=1}^{n} x_i y_i - x_0 \sum_{j=1}^{n+m} y_j$ that appears in (22a). However, upon complementing x_0, a simple application of Luedtke et al. [69], Theorem 8, shows that this is not the case (cf. §1.5.1, Observation 1.2).

We next compare the strengths of the two relaxations $\mathcal{M}(\mathcal{P})$ and $\mathcal{SM}(\mathcal{P})$. The comparison depends upon the values of the capacity u_0 and implied capacity $\sum_{j>0} u_j$.

Proposition 2.4. *The strengths of the two polyhedral relaxations, $\mathcal{M}(\mathcal{P})$ and $\mathcal{SM}(\mathcal{P})$, can be compared as follows.*

1. *If $u_i < u_0$ for some i, then $\mathrm{Proj}_{x,y}\, \mathcal{M}(\mathcal{P}) \nsubseteq \mathrm{Proj}_{x,y}\, \mathcal{SM}(\mathcal{P})$.*

2. *If $\sum_{j=1}^{n+m} u_j \le u_0$, then $\mathrm{Proj}_{x,y}\, \mathcal{SM}(\mathcal{P}) \subseteq \mathrm{Proj}_{x,y}\, \mathcal{M}(\mathcal{P})$.*

3. *If $\sum_{j=1}^{n+m} u_j > u_0$ and $\exists \mathcal{T} \subseteq \{n+1, \dots, n+m\}$ such that*

   (a) $\sum_{j=1}^{n} u_j + \sum_{j \in \mathcal{T}} a_{j-n} u_j > u_0$, or else

   (b) $\sum_{j=1}^{n} u_j + \sum_{j=n+1}^{n+m} a_{j-n} u_j < u_0$ and $\sum_{j \in \mathcal{T}} (1 - a_{j-n}) u_j > u_0 \min_{j \in \mathcal{T}} (1 - a_{j-n})$,

   then $\mathrm{Proj}_{x,y}\, \mathcal{SM}(\mathcal{P}) \nsubseteq \mathrm{Proj}_{x,y}\, \mathcal{M}(\mathcal{P})$.

4. *If $\sum_{j=1}^{n+m} u_j > u_0$ and for all $j = 1, \dots, n+m, u_j \ge u_0$, and hence $u_j = u_0$ due to Assumption 2.1, then $\mathrm{Proj}_{x,y}\, \mathcal{M}(\mathcal{P}) \subseteq \mathrm{Proj}_{x,y}\, \mathcal{SM}(\mathcal{P})$.*

Proof. First suppose that $u_1 < u_0$. Consider a point $(x, y, w) \in \mathcal{M}(\mathcal{P})$ such that $x_k = 0$, for all $k = 2, \dots, n$, $y_k = 0$, for all $k = 2, \dots, n+m$, and $w_0 = u_0 x_0$. Further, let

$x_0 = u_1/u_0, x_1 = 1, y_1 = u_1$. Hence, $w_0 = y_1 = u_1 = w_1$. Since $y_k = 0, \forall k \geq 2$, then it must be that $w_{0k} = 0, \forall k \geq 2$, and $w_{01} = w_1 = u_1$. Now, $u_1 < u_0$ implies $x_0 < 1$ and hence $w_{01} = u_1 > u_1 x_0$, violating its concave envelope. Thus this chosen point (x, y) cannot be in $\text{Proj}_{x,y}\, \mathcal{SM}(\mathcal{P})$.

Now let $\sum_{j=1}^{n+m} u_j \leq u_0$. Take a point $(x, y, w) \in \mathcal{SM}(\mathcal{P})$. Then it must be that

$$\max\{0, \sum_{j>0} u_j(x_0 - 1) + \sum_j y_j\} \;\leq\; \sum_{j>0} w_{0j} \;\leq\; \min\{\sum_{j>0} u_j x_0, \sum_j y_j\}.$$

Since $u_0 \geq \sum_{j>0} u_j$ and $x_0 \leq 1$, it follows that $\sum_{j>0} u_j(x_0 - 1) \geq u_0(x_0 - 1)$ and $\sum_{j>0} u_j x_0 \leq u_0 x_0$. Setting $w_0 = \sum_{j>0} w_{0j}$ implies that this point is in $\mathcal{M}(\mathcal{P})$.

Now suppose that $\sum_{j=1}^{n+m} u_j > u_0$ and $\mathcal{T} \subseteq \{n+1, \ldots, n+m\}$. Construct a point in $(x, y, w) \in \mathcal{SM}(\mathcal{P})$ as follows. Set $w_{0j} = y_j = u_j x_0$, for all $j = 1, \ldots, n$, $w_{0j} = a_{j-n} u_j x_0, y_j = u_j x_0$, for all $j \in \mathcal{T}$, and $w_{0j} = y_j = 0$, for all $j \in \{n+1, \ldots, n+m\} \setminus \mathcal{T}$. Also, set $x_i = 1$ and hence $w_i = y_i = u_i x_0$, for all $i = 1, \ldots, n$. The value of x_0 is

$$x_0 = \min\left\{ \frac{u_0}{\sum_{j=1}^{n} u_j + \sum_{j \in \mathcal{T}} u_j}, \; \min_{j \in \mathcal{T}} \frac{1}{2 - a_{j-n}} \right\}.$$

By construction, $x_0 \in (0, 1]$ and this point belongs to $\mathcal{SM}(\mathcal{P})$ as discussed next. The capacity constraint is $\sum_{j>0} y_j = x_0[\sum_{j=1}^{n} u_j + \sum_{j \in \mathcal{T}} u_j] \leq u_0$. The concave envelopes of w_{0j} are satisfied because $a_{j-n} \leq 1$. For the nontrivial convex envelope, note that for $j = 1, \ldots, n, w_{0j} = u_j x_0 \geq u_j x_0 + y_j - u_j$ and for $j \in \mathcal{T}$,

$$\begin{aligned} a_{j-n} u_j x_0 &\geq u_j x_0 + u_j x_0 - u_j \\ \iff x_0 &\leq \frac{1}{2 - a_{j-n}}. \end{aligned}$$

To construct a point in $\mathcal{M}(\mathcal{P})$, since $w_i = y_i, \forall i = 1, \ldots, n$, we must have

$$w_0 = \sum_{j>0} w_{0j} = x_0 \left[\sum_{j=1}^{n} u_j + \sum_{j \in \mathcal{T}} a_{j-n} u_j \right].$$

Consider the two conditions for $\mathcal{T}$.

$\sum_{j=1}^{n} u_j + \sum_{j \in \mathcal{T}} a_{j-n} u_j > u_0$ **:** In this case, the concave envelope $w_0 \leq u_0 x_0$ is violated.

Else $\sum_{j\in\mathcal{T}}(1-a_{j-n})u_j > u_0 \min_{j\in\mathcal{T}}(1-a_{j-n})$ **:** We also assume in this case that $\sum_{j=1}^{n} u_j + \sum_{j=n+1}^{n+m} a_{j-n}u_j < u_0$. Then,

$$\begin{aligned} \sum_{j=1}^{n} u_j + \sum_{j\in\mathcal{T}} a_{j-n}u_j &< u_0 \\ \iff \sum_{j=1}^{n} u_j + \sum_{j\in\mathcal{T}} u_j &< u_0 + \sum_{j\in\mathcal{T}}(1-a_{j-n})u_j \\ \iff \frac{u_0}{\sum_{j=1}^{n} u_j + \sum_{j\in\mathcal{T}} u_j} &> \frac{u_0}{u_0 + \sum_{j\in\mathcal{T}}(1-a_{j-n})u_j}. \end{aligned}$$

Also, for some $t \in \mathcal{T}$,

$$\begin{aligned} \sum_{j\in\mathcal{T}}(1-a_{j-n})u_j &> u_0(1-a_{t-n}) \\ \iff \frac{1}{2-a_{t-n}} &> \frac{u_0}{u_0 + \sum_{j\in\mathcal{T}}(1-a_{j-n})u_j}. \end{aligned}$$

Hence, $x_0 > u_0/(u_0 + \sum_{j\in\mathcal{T}}(1-a_{j-n})u_j)$. Now, the convex envelope $w_0 \geq u_0x_0 + \sum_{j=1}^{n+m} y_j - u_0$ is violated as follows.

$$\begin{aligned} w_0 &< u_0x_0 + \sum_{j=1}^{n+m} y_j - u_0 \\ \iff x_0\left[\sum_{j=1}^{n} u_j + \sum_{j\in\mathcal{T}} a_{j-n}u_j\right] &< u_0(x_0-1) + x_0\left[\sum_{j=1}^{n} u_j + \sum_{j\in\mathcal{T}} u_j\right] \\ \iff x_0 &> \frac{u_0}{u_0 + \sum_{j\in\mathcal{T}}(1-a_{j-n})u_j}. \end{aligned}$$

Hence, this point cannot belong to $\mathcal{M}(\mathcal{P})$ under the above two conditions on $\mathcal{T}$.

For the final case, starting with a point in $\mathcal{M}(\mathcal{P})$, we construct a point in $\mathcal{SM}(\mathcal{P})$ by setting $w_{0j} = w_0 y_j / \sum_{k=1}^{n+m} y_k$ for all $j = 1, \ldots, n+m$. By construction, we have that $\sum_{j=1}^{n+m} w_{0j} = w_0$. We now verify that this point satisfies the McCormick envelopes for $w_{0j}, \forall j$. Clearly, $w_0 \geq 0$ and $y \geq \mathbf{0}$ implies $w_{0j} \geq 0$. Now,

$$\begin{aligned} w_{0j} = \frac{w_0\, y_j}{\sum_k y_k} &\geq \left[u_0x_0 + \sum_k y_k - x_0\right]\frac{y_j}{\sum_k y_k} \\ &= u_0(x_0-1)\frac{y_j}{\sum_k y_k} + y_j \\ &\geq u_0(x_0-1) + y_j \qquad \text{since } y_j \leq \sum_k y_k,\ x_0 \leq 1 \\ &= u_j(x_0-1) + y_j \qquad \text{since } u_0 = u_j. \end{aligned}$$

For the concave envelopes, $y_j \leq \sum_k y_k, w_0 \geq 0$ implies $w_{0j} \leq w_0 \leq u_0 x_0 = u_j x_0$ and $w_0 \leq \sum_k y_k, y_j \geq 0$ implies $w_{0j} \leq y_j$. □

We now show that the McCormick envelopes can lead to strong relaxations under certain assumptions. Recall the set $\mathcal{Q}$ defined in (21). Under the assumptions of Proposition 2.1, we proved that the convex hull of $\mathcal{Q}$ is a polyhedral set. We now show that if $a = \mathbf{e}$, then this convex hull is given by the McCormick relaxation $\mathcal{M}(\mathcal{Q})$. The main ingredient of this proof is the following lemma.

Lemma 2.2. *Let Ω and Ψ be two matrices defined as*

$$\Omega := \begin{bmatrix} 0 & 0 & -1 \\ -1 & 0 & 1 \\ 0 & -1 & 1 \\ 1 & 1 & -1 \end{bmatrix}, \qquad \Psi := \begin{bmatrix} \Omega & & & & & & & \\ & \Omega & & & & & & \\ & & \ddots & & & & & \\ & & & \ddots & & & & \\ & & & & \ddots & & & \\ & & & & & \Omega & & \\ \mathbf{e}_3 & \dots & \mathbf{e}_3 & -\mathbf{e}_3 & \dots & -\mathbf{e}_3 & \mathbf{e} & -\mathbf{e} \\ -\mathbf{e}_3 & \dots & -\mathbf{e}_3 & \mathbf{e}_3 & \dots & \mathbf{e}_3 & -\mathbf{e} & \mathbf{e} \end{bmatrix},$$

where Ψ is a $(4n+2) \times (3n+m)$ matrix with n diagonal blocks of Ω, for some positive integers n, m. Then, Ψ is a totally unimodular (TU) matrix.

Proof. To prove that Ψ is TU, we will show that for any subset of rows $T \subseteq \{1, \dots, 4n+2\}$, there exists a partition (T_1, T_2) such that for every column j of Ψ, we have

$$\left| \sum_{t \in T_1} \Psi_{tj} - \sum_{t \in T_2} \Psi_{tj} \right| \leq 1 \tag{27}$$

For $i = 1, \dots, n$, denote $R_i := \{4i+1, 4i+2, 4i+3, 4i+4\}$ as the rows of Ψ that have the matrix Ω on its i^{th} block diagonal. Let $R_i^+ := \{4i+2, 4i+3\} \subset R_i$ be the rows with a $+1$ coefficient in the last column of Ω and let $R_i^- := R_i \setminus R_i^+$. We consider two different cases based on the composition of T.

Case 1 $|T \cap \{4n+1, 4n+2\}| \in \{0, 2\}$

Since Ω is TU, we can partition its rows belonging to T into two subsets. Let $(T_1^{(i)}, T_2^{(i)})$ be such a partition of $T \cap R_i$. Set $T_1 = \cup_{i=1}^{n} T_1^{(i)}$ and $T_2 = \Big(T \cap \{4n+1, 4n+2\}\Big) \bigcup \Big(\cup_{i=1}^{n} T_2^{(i)}\Big)$.

Choose some column j of the matrix Ψ. If this column has nonzeros in only the last two rows, then it is trivial to verify (27). Now let i^* be such that the j^{th} column corresponds to i^*th block of Ω. Given the structure of Ψ and observing that $\sum_{t \in T \cap \{4n+1, 4n+2\}} \Psi_{tj} = 0$, we conclude that

$$\sum_{t \in T_1} \Psi_{tj} - \sum_{t \in T_2} \Psi_{tj} = \sum_{t \in T_1^{(i^*)}} \Psi_{tj} - \sum_{t \in T_2^{(i^*)}} \Psi_{tj}$$

Since $(T_1^{(i^*)}, T_2^{(i^*)})$ is a partition of $T \cap R_i$, which forms a subset of rows of Ω, and Ω is TU, the difference on the right hand side is no greater than 1 in absolute value.

Case 2 $|T \cap \{4n+1, 4n+2\}| = 1$

Suppose that $4n+1 \in T, 4n+2 \notin T$. Initialize $T_1 = \emptyset$ and $T_2 = \{4n+1\}$. For simplicity, let $n = n_1 + n_2$, where n_1, n_2 are such that the $4n+1^{th}$ row of Ψ contains n_1 consecutive $\mathbf{e}_3$'s and n_2 consecutive $-\mathbf{e}_3$'s. Then let $I = \{1, \ldots, n_1\}$ and suppose $i \in I$. Consider the following two cases for the composition of T.

$R_i^+ \subseteq T$ **:** Set $T_1 = T_1 \cup (T \cap R_i)$.

$R_i^+ \nsubseteq T$ **:** First suppose that $T \cap R_i^+ = \emptyset$. Then $T_2 = T \cap R_i^-$. Otherwise T contains exactly one row from R_i^+, say l, since $R_i^+ \nsubseteq T$. Here we set $T_1 = T_1 \cup \{l, r\}$ and $T_2 = T_2 \cup (T \cap R_i^- \setminus \{r\})$ for some row $r \in T \cap R_i^-$.

We now argue that the above construction of the subsets T_1 and T_2 satisfies (27). Let $T_1^{(i)}$ and $T_2^{(i)}$ be the set of rows from the i^{th} block of Ψ that are in T_1 and T_2, respectively. Thus, $T_1 = \cup_i T_1^{(i)}$ and $T_2 = (\cup_i T_2^{(i)}) \bigcup \{4n+1\}$.

Choose a column j of the matrix Ψ and let $i \in I$ be such that the j^{th} column corresponds to ith block of Ω. Observe that $\Psi_{4n+1,j} \in \{0, 1\}$ and

$$\sum_{t \in T_1} \Psi_{tj} - \sum_{t \in T_2} \Psi_{tj} = \sum_{t \in T_1^{(i)}} \Psi_{tj} - \sum_{t \in T_2^{(i)}} \Psi_{tj} - \Psi_{4n+1,j} \tag{28}$$

$R_i^+ \subseteq T$ **:** Our construction yields $T_1^{(i)} = T \cap R_i$ and $T_2^{(i)} = \emptyset$. Then, we have $\sum_{t \in T \cap R_i} \Psi_{tj} = \sum_{t \in R_i^+} \Psi_{tj} + \sum_{t \in T \cap R_i^-} \Psi_{tj}$. From the definition of Ω, it is easy to see that $\sum_{t \in R_i^+} \Psi_{tj}$ is either -1 or +2. Adding more rows from $T \cap R_i^-$ and subtracting $\Psi_{4n+1,j}$ in equation (28) maintains the column sum to less than or equal to 1 in absolute value.

$R_i^+ \nsubseteq T$ **:** In this case, we show how to select an appropriate row r such that (27) is satisfied. If $4i+4 \in T$, then set $r = 4i+4$. Else if $4i+1 \in T$, then $r = 4i+1$. If neither of these conditions hold, then $\{r\} = \{\emptyset\}$. It is easily verified that these choices for r satisfy (27).

For $i \in \{1, \ldots, n\} \setminus I$, we simply interchange R_i^+ and R_i^-. The other case when $4n+1 \notin T, 4n+2 \in T$ can be addressed by interchanging the initializations of T_1 and T_2.

□

The above result helps us in the following way. Observe that the matrix Ω is the coefficient matrix for the convex and concave envelopes of a individual bilinear term $w = xy$, with the columns being sorted as (x, y, w). Then Ψ becomes the coefficient matrix for the McCormick relaxation of the set $\mathcal{Q}$ with $a = \mathbf{e}$, all lower bounds zero and all upper bounds equal to one. Lemma 2.2 implies that as long as the right hand side is integral, then the extreme points of the McCormick relaxation $\mathcal{M}(\mathcal{Q})$ are $\{0, 1\}$ in each element. Since McCormick envelopes are exact at the bounds, it follows that these extreme points satisfy $w_i = x_i y_i, \forall i$, and hence belong to $\mathcal{Q}$. This immediately implies the next corollary.

Corollary 2.3. *Consider the assumptions of Proposition 2.1 and further assume that* $U = 1$ *and* $a = \mathbf{e}$. *Then,* $\text{conv}(\mathcal{Q}) = \text{Proj}_{x,y} \mathcal{M}(\mathcal{Q})$.

2.4 Disjunctive formulation

In this section, we study restrictions of $\mathcal{P}$ in $\Re^2$ obtained by fixing the remaining variables. Using these restrictions, we represent $\text{conv}(\mathcal{P})$ as the convex hull of a union of a finite number of conic quadratic sets. For each restriction, we also present a family of polyhedral

relaxations using tangent and secant inequalities. Convexifying the union of these polyhedra gives a polyhedral relaxation of $\text{conv}(\mathcal{P})$. We show that this disjunctive relaxation is stronger than both the McCormick relaxations from §2.3.

2.4.1 Restrictions using extreme values

We construct restrictions of $\mathcal{P}$ using its extreme point characterization from Theorem 2.1. Note that at any extreme point of $\text{conv}(\mathcal{P})$, there is only one index j such that y_j takes all values in some bounded interval. We also note that due to the capacity constraint (22b), the remaining variables $\{y_i\}_{i \neq j}$, cannot be arbitrarily fixed to their upper bounds. To formalize this notion, we introduce the following definition.

Definition 2.2. A subset $C \subseteq \{1, \ldots, n+m\}$ is said to be *independent* if $\sum_{i \in C} u_i \leq u_0$. C is strictly independent if the inequality is strict.

Let $N := \{1, \ldots, n+m\}$ and $I := \{1, \ldots, n\}$. Given a $j \in N$ and subsets $N_1 \subseteq I$ and $N_2 \subseteq N \setminus j$ such that N_2 is strictly independent, let $N_1^- \subseteq I \setminus N_1$ and $N_2^- \subseteq N \setminus (N_2 \cup j)$. We consider restrictions of $\mathcal{P}$ defined as follows.

Definition 2.3. $\mathcal{F}(N_1^-, N_1, N_2^-, N_2)$ is a restriction of $\mathcal{P}$ obtained by fixing variables as:

1. x_i, for all $i \in N_1^-$, are fixed to 0,
2. x_i, for all $i \in N_1$, are fixed to 1,
3. y_i, for all $i \in N_2^-$, are fixed to 0, and
4. y_i, for all $i \in N_2$, are fixed to u_i.

We denote the fixed values of these variables by $(\bar{x}, \bar{y})$. Thus, $\bar{x}_i = 1, \forall i \in N_1, \bar{x}_i = 0, \forall i \in N_1^-, \bar{y}_i = u_i, \forall i \in N_2$, and $\bar{y}_i = 0, \forall i \in N_2^-$.

Whenever $N_1^- = I \setminus N_1$ and $N_2^- = N \setminus (N_2 \cup j)$, for brevity, we will denote

$$\mathcal{F}(N_1, N_2) := \mathcal{F}(I \setminus N_1, N_1, N \setminus (N_2 \cup j), N_2).$$

Thus $\mathcal{F}(N_1, N_2)$ is a restriction of $\mathcal{P}$ in the (x_0, y_j)-space. □

Let $\bar{p}$ and $\bar{q}$ be defined as

$$\bar{p} := \sum_{i \in N_2} u_i, \qquad \bar{q} := \sum_{i \in N_1 \cap N_2} u_i + \sum_{i \in (N \setminus I) \cap N_2} a_{i-n} u_i. \tag{29}$$

Since $a_{i-n} \in [0,1], \forall i$, we have $\bar{q} \leq \bar{p}$. For any $k \notin N_2$, define

$$\mu_k := \min\{u_k, u_0 - \bar{p}\}. \tag{30}$$

Since N_2 is strictly independent and $u_j > 0$ by Assumption 2.1, it follows that $\mu_j > 0$. Theorem 2.1 implies that there exist extreme points satisfying $y_j \in [0, \mu_j]$. The set $\mathcal{F}(N_1, N_2)$ can be explicitly described as

$$\mathcal{F}(N_1, N_2) = \{(x_0, y_j) \in \Re^2_+ : \ -\lambda_j y_j + x_0 y_j + \bar{p} x_0 = \bar{q}, \ x_0 \leq 1, \ y_j \leq \mu_j\}, \tag{31}$$

where λ_j is a parameter defined as

$$\lambda_j := \begin{cases} 1, & \text{if } j \in N_1 \\ 0, & \text{if } j \in N_1^- \\ a_{j-n}, & \text{else } j \in N \setminus I. \end{cases} \tag{32}$$

As seen in (31), a bilinear equality constraint defines $\mathcal{F}(N_1, N_2)$. This description can be further simplified depending on the values of $\bar{p}$ and $\bar{q}$. We next discuss this for each case.

$j \in N_1$ **:** If $\bar{p} > \bar{q}$, then $x_0 < 1$. Hence, we can rewrite the bilinear equality in (31) as $y_j = \frac{\bar{p}x_0 - \bar{q}}{1 - x_0}$. In this case,

$$\mathcal{F}(N_1, N_2) = \left\{ (x_0, y_j) \in \Re^2_+ : y_j = \frac{\bar{p}x_0 - \bar{q}}{1 - x_0}, \ y_j \leq \mu_j, \ x_0 \in \left[\frac{\bar{q}}{\bar{p}}, \frac{\bar{q} + \mu_j}{\bar{p} + \mu_j} \right] \right\}.$$

Otherwise if $\bar{p} = \bar{q} = 0$, then $\mathcal{F}(N_1, N_2)$ is defined by $y_j(1 - x_0) = 0$, which implies that the feasible set is $\{1 \times [0, u_0 - \bar{p}]\} \cup \{[0,1] \times 0\}$. Finally, if $\bar{p} = \bar{q} > 0$, then $x_0 = 1$ and hence $\mathcal{F}(N_1, N_2) = \{1 \times [0, u_0 - \bar{p}]\}$.

$j \in N_1^-$ **:** If $\bar{q} > 0$, then $x_0 > 0$ and hence the bilinear equality can be reformulated as $y_j = \frac{\bar{q} - \bar{p}x_0}{x_0}$. Hence,

$$\mathcal{F}(N_1, N_2) = \left\{ (x_0, y_j) \in \Re^2_+ : y_j = \frac{\bar{q} - \bar{p}x_0}{x_0}, \ y_j \leq \mu_j, \ x_0 \in \left[\frac{\bar{q}}{\bar{p} + \mu_j}, \frac{\bar{q}}{\bar{p}} \right] \right\}.$$

Otherwise if $\bar{p} = \bar{q} = 0$, then the feasible set is $\{0 \times [0, u_0 - \bar{p}]\} \cup \{[0,1] \times 0\}$. Else $\bar{p} > \bar{q} = 0$ and the set is $\{0 \times [0, u_0 - \bar{p}]\}$.

$j \in N \setminus I$ **:** First suppose that $\bar{p} > \bar{q}$. We consider three subcases. If $\bar{q}/\bar{p} < a_{j-n}$, then $x_0 \geq a_{j-n}$ either implies $\bar{q} = a_{j-n}\bar{p}$ or $y_j < 0$, neither of which is possible. Hence it must be that $x_0 < a_{j-n}$ and the bilinear constraint becomes $y_j = \frac{\bar{p}x_0 - \bar{q}}{a_{j-n} - x_0}$, giving us

$$\mathcal{F}(N_1, N_2) = \left\{ (x_0, y_j) \in \Re^2_+ \colon y_j = \frac{\bar{p}x_0 - \bar{q}}{a_{j-n} - x_0},\ y_j \leq \mu_j,\ x_0 \in \left[\frac{\bar{q}}{\bar{p}},\ \frac{\bar{q} + a_{j-n}\mu_j}{\bar{p} + \mu_j} \right] \right\}.$$

Otherwise if $\bar{q}/\bar{p} > a_{j-n}$, then the argument is similar to the previous case with the only difference being that $x_0 > a_{j-n}$ and

$$\mathcal{F}(N_1, N_2) = \left\{ (x_0, y_j) \in \Re^2_+ \colon y_j = \frac{\bar{p}x_0 - \bar{q}}{a_{j-n} - x_0},\ y_j \leq \mu_j,\ x_0 \in \left[\frac{\bar{q} + a_{j-n}\mu_j}{\bar{p} + \mu_j},\ \frac{\bar{q}}{\bar{p}} \right] \right\}.$$

Else $\bar{q}/\bar{p} = a_{j-n}$ and it is easily verified that the feasible set is $\{a_{j-n} \times [0, \mu_j]\}$.

Now suppose that $\bar{p} = \bar{q} > 0$. The case $x_0 < a_{j-n}$ yields an empty feasible set since $\bar{p} > 0$. If $x_0 = a_{j-n}$, then a_{j-n} must be 1 for a nonempty feasible set, which is given by $\{1 \times [0, \mu_j]\}$. Else $x_0 > a_{j-n}$ and

$$\mathcal{F}(N_1, N_2) = \left\{ (x_0, y_j) \in \Re^2_+ \colon y_j = \frac{\bar{p}x_0 - \bar{p}}{a_{j-n} - x_0},\ y_j \leq \mu_j,\ x_0 \in \left[\frac{\bar{p} + a_{j-n}\mu_j}{\bar{p} + \mu_j},\ 1 \right] \right\}.$$

Finally suppose that $\bar{p} = \bar{q} = 0$. Then (31) implies the feasible set is $(a_{j-n} - x_0)y_j = 0$, which can be equivalently written as the set $\{[0, 1] \times 0\} \cup \{a_{j-n} \times [0, \mu_j]\}$.

The above discussion gives us necessary and sufficient conditions to determine the polyhedrality of the convex hull of $\mathcal{F}(N_1, N_2)$. First, consider the following definition.

Definition 2.4. $\psi_j \colon [0, 1] \mapsto \Re \cup \{+\infty\}$ such that $\psi_j(x_0) := \dfrac{\bar{p}x_0 - \bar{q}}{\lambda_j - x_0}$ for $x_0 \neq \lambda_j$ and $+\infty$ otherwise.

Proposition 2.5. $\mathrm{conv}(\mathcal{F}(N_1, N_2))$ *is non-polyhedral if and only if one of the following three conditions is satisfied,*

1. $j \in N_1$ *and* $\bar{p} > \bar{q}$, *or*
2. $j \in N_1^-$ *and* $\bar{q} > 0$, *or*
3. $j \in N \setminus I$, *and either*
   (a) $\bar{q}/\bar{p} \in [0, a_{j-n})$, *or*

(b) $\bar{q}/\bar{p} \in (a_{j-n}, 1)$, *or*

(c) $\bar{p} = \bar{q} > 0$ *with* $a_{j-n} \in [0, 1)$.

If any one these conditions is satisfied, then we can represent $\mathcal{F}(N_1, N_2)$ *as*

$$\mathcal{F}(N_1, N_2) = \left\{ (x_0, y_j) \in \Re^2_+ \colon y_j = \psi_j(x_0),\ y_j \leq \mu_j,\ x_0 \in \left[\frac{\bar{q}}{\bar{p}},\ \frac{\bar{q} + \lambda_j \mu_j}{\bar{p} + \mu_j} \right] \right\},$$

where an interval $[\ell, \vartheta]$ *is regarded as* $[\vartheta, \ell]$ *for* $\vartheta < \ell$.

Corollary 2.4. $\mathrm{conv}(\mathcal{F}(N_1, N_2))$ *is polyhedral if and only if*

1. $\bar{p} = \bar{q} = 0$, *or*

2. $\bar{p} > 0$ *and* $\bar{q} = \bar{p}\lambda_j$.

Definition 2.5. A restriction $\mathcal{F}(N_1, N_2)$ is said to be *nontrivial* if its convex hull is non-polyhedral; equivalently if any one of the conditions of Proposition 2.5 is satisfied. Otherwise, $\mathcal{F}(N_1, N_2)$ is trivial.

Definition 2.6. A nontrivial $\mathcal{F}(N_1, N_2)$ is said to be *right-leaning* if (1) or (3a) from Proposition 2.5 are satisfied. Otherwise, it is said to be *left-leaning* when (2), (3b), or (3c) are satisfied. See figures 5(a) and 5(b), respectively. The set of all right- and left-leaning restrictions is $\mathfrak{F}^{\searrow}$ and $\mathfrak{F}^{\swarrow}$, respectively.

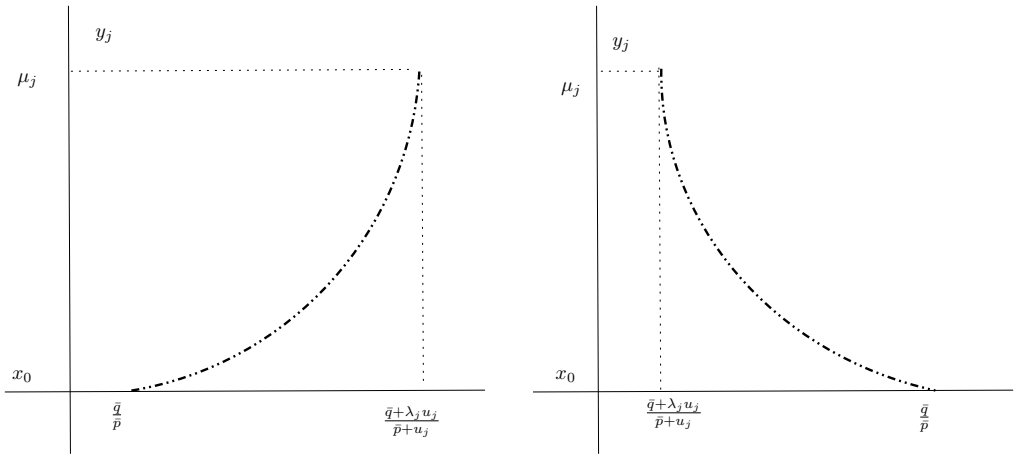
The trivial and nontrivial restrictions of $\mathcal{P}$ are illustrated in Figure 5.

Given a restriction $\mathcal{F}(N_1, N_2)$, let $\mathcal{D}$ be the range of values taken by the variable x_0, i.e. $\mathcal{D} = \{x_0 \colon (x_0, y_j) \in \mathcal{F}(N_1, N_2)\}$. For convenience, denote $\ell = \bar{q}/\bar{p}$ and $\vartheta = (\bar{q} + \lambda_j \mu_j)/(\bar{p} + \mu_j)$. For a nontrivial $\mathcal{F}(N_1, N_2)$, we have $\mathcal{D} = [\ell, \vartheta]$ with the understanding that $[\ell, \vartheta]$ is regarded as $[\vartheta, \ell]$ if $\vartheta < \ell$. The proof of Proposition 2.5 implies that $\vartheta < \ell$ if and only if the restriction is left-leaning.

Observation 2.4. *The following statements are equivalent.*

1. $\mathcal{F}(N_1, N_2) \in \mathfrak{F}^{\swarrow}$.

2. $\vartheta < \ell$.



(a) Right-leaning $\mathcal{F}(N_1, N_2)$

(b) Left-leaning $\mathcal{F}(N_1, N_2)$

(c) Trivial $\mathcal{F}(N_1, N_2)$: $\bar{q} = \bar{p} = 0$

(d) Trivial $\mathcal{F}(N_1, N_2)$: $\bar{q} = \bar{p}\lambda_j$, $\bar{p} > 0$

Figure 5: Restrictions of $\mathcal{P}$ using extremal values.

3. $\bar{q}/\bar{p} > \lambda_j$.

The discussion preceding Proposition 2.5 implies the next observation.

Observation 2.5. *Let* $\mathcal{F}(N_1, N_2)$ *be nontrivial. Then either* $x_0 < \lambda_j$ *or* $x_0 > \lambda_j$ *for every* $(x_0, y_j) \in \mathcal{F}(N_1, N_2)$ *and*

$$x_0 < \lambda_j \iff \bar{p}\lambda_j - \bar{q} > 0 \iff \mathcal{F}(N_1, N_2) \in \mathfrak{F}^{\searrow}.$$

Hence the equality $y_j = \psi_j(x_0)$ *is well-defined and* $\mathcal{D} \subset [0, 1]$ *is such that either* $\mathcal{D} \subset [0, \lambda_j)$ *or* $\mathcal{D} \subset (\lambda_j, 1]$.

We next show that $\psi_j(\cdot)$ is convex over $\mathcal{D}$ and is the only nonlinear convex inequality

defining the convex hull of $\mathcal{F}(N_1, N_2)$. In particular, Lemma 2.3 shows that the convex hull of $\mathcal{F}(N_1, N_2)$ is given by the intersection of the epigraph of $\psi_j(\cdot)$ and the secant inequality joining the two endpoints $(\ell, 0)$ and (ϑ, μ_j).

Lemma 2.3. *Suppose that $\mathcal{F}(N_1, N_2)$ is nontrivial. The function $\psi_j(\cdot)$ is conic quadratic representable (CQr), and hence convex, over $\mathcal{D}$. Furthermore,*

$$\operatorname{conv}(\mathcal{F}(N_1, N_2)) = \left\{(x_0, y_j) \in \Re^2_+ : y_j \geq \psi_j(x_0),\ -\mu_j x_0 + (\vartheta - \ell) y_j \simeq -\mu_j \ell\right\},$$

where $\simeq$ is $\leq$ if $\ell < \vartheta$ and $\simeq$ is $\geq$ if $\ell > \vartheta$.

Proof. $x_0 \in \mathcal{D}$ implies $x_0 \neq \lambda_j$. Assume that $\mathcal{D} \subset [0, \lambda_j)$ and hence $x_0 < \lambda_j$. We can reformulate $\psi_j(\cdot)$ as

$$\psi_j(x_0) = -\bar{p} + \frac{\bar{p}\lambda_j - \bar{q}}{\lambda_j - x_0}.$$

Hence the epigraph of $\psi_j(\cdot)$ can be written as a conic quadratic set as follows.

$$\begin{aligned} y_j \geq \psi_j(x_0) \quad &\iff \quad y_j \geq -\bar{p} + \frac{\bar{p}\lambda_j - \bar{q}}{\lambda_j - x_0} \\ &\iff \quad (y_j + \bar{p})(\lambda_j - x_0) \geq \bar{p}\lambda_j - \bar{q} \\ &\iff \quad \left\| (y_j + \bar{p} - \lambda_j + x_0,\ 2\sqrt{\bar{p}\lambda_j - \bar{q}}) \right\|_2 \leq y_j + \bar{p} + \lambda_j - x_0, \end{aligned}$$

where in the last equivalence we use the fact that $y_j + \bar{p} + \lambda_j - x_0 \geq 0$. The argument for $x_0 > \lambda_j$ is similar by noting from Observation 2.5 that $x_0 > \lambda_j \iff \bar{p}\lambda_j - \bar{q} < 0$. Thus, $\psi_j(\cdot)$ is a conic quadratic representable function and is hence convex over $\mathcal{D}$. It follows that $\mathcal{F}(N_1, N_2)$ is the set of all the points lying on a convex curve over a bounded interval in $\Re_+$ and hence its convex hull is given by the epigraph of the convex function and a secant inequality. □

2.4.2 High-dimensional representations

The restriction $\mathcal{F}(N_1, N_2)$ was constructed in the (x_0, y_j)-space by fixing the remaining variables at one of their extreme point values, as characterized by Theorem 2.1. Since

$\text{conv}(\mathcal{P})$ is given by all possible convex combinations of its extreme points, it follows that

$$\begin{aligned} \text{conv}(\mathcal{P}) &= \text{conv} \bigcup_{\substack{j \in N \\ N_1 \subseteq I}} \bigcup_{\substack{N_2 \subseteq N \setminus j: \\ N_2 \in \mathcal{C}}} \mathcal{F}(N_1, N_2) \\ &= \text{conv} \bigcup_{\substack{j \in N \\ N_1 \subseteq I}} \bigcup_{\substack{N_2 \subseteq N \setminus j: \\ N_2 \in \mathcal{C}}} \text{conv}(\mathcal{F}(N_1, N_2)) \end{aligned} \tag{33}$$

where $\mathcal{C}$ is the collection of all strictly independent sets (cf. Definition 2.2). If the convex hull of $\mathcal{F}(N_1, N_2)$ is non-polyhedral, then it is given by Lemma 2.3, otherwise it is polyhedral and can be easily obtained from the discussion preceding Proposition 2.5. Then, Lemma 2.3 implies that $\text{conv}(\mathcal{P})$ is the convex hull of union of finitely many bounded conic and polyhedral sets. A higher-dimensional formulation for the convex hull of $\mathcal{P}$ can then be obtained using convex disjunctive programming techniques (cf. Ceria and Soares [30]).

Corollary 2.5. $\text{conv}(\mathcal{P})$ *is CQr.*

Proof. By Lemma 2.3, $\text{conv}(\mathcal{F}(N_1, N_2))$ is CQr when $\mathcal{F}(N_1, N_2)$ is nontrivial. Otherwise, $\mathcal{F}(N_1, N_2)$ is trivial and hence its convex hull is polyhedral, which is also CQr. Since $\text{conv}(\mathcal{P})$ is closed and is the convex hull of union of finitely many closed CQr sets (cf. (33)), the result follows from Ben-Tal and Nemirovski [24], Proposition 2.3.5. □

Since $\psi_j(\cdot)$ is a convex function, the gradient inequalities are valid to its epigraph. This allows constructing a polyhedral relaxation of $\mathcal{F}(N_1, N_2)$ by replacing the epigraph of $\psi_j(\cdot)$ with gradient inequalities formed with respect to a finite number of (say k) points on the epigraph of $\psi_j(\cdot)$. (If $\mathcal{F}(N_1, N_2)$ is trivial, then there is no need for a gradient relaxation). The gradients can be constructed at arbitrary points on the epigraph. However, to avoid getting a weak relaxation, we henceforth enforce that $k \geq 2$ and the two endpoints of $\mathcal{D} = [\ell, \vartheta]$ are always selected. Let $\hat{x}_0^t \in \mathcal{D}$, for $t = 1, \ldots, k$, be the points (sorted in increasing order) with respect to which gradient inequalities are generated. Denote this k-gradient polyhedral relaxation of $\mathcal{F}(N_1, N_2)$ by $\mathcal{T}_k(N_1, N_2)$.

$$\begin{aligned} \mathcal{T}_k(N_1, N_2) = \{(x_0, y_j) \colon\ & y_j \geq \psi_j(\hat{x}_0^t) + \frac{\bar{p}\lambda_j - \bar{q}}{(\lambda_j - \hat{x}_0^t)^2} \left(x_0 - \hat{x}_0^t\right), \quad t = 1, \ldots, k \\ & -\mu_j x_0 + (\vartheta - \ell) y_j \ \simeq\ -\mu_j \ell \}. \end{aligned} \tag{34}$$

Taking unions over all such relaxations produces a polyhedral relaxation of $\text{conv}(\mathcal{P})$,

$$\text{conv}(\mathcal{P}) \subset \mathcal{T}_k := \text{conv} \bigcup_{j \in N} \bigcup_{\substack{N_1 \subseteq I \\ N_2 \subseteq N \setminus j}} \mathcal{T}_k(N_1, N_2), \tag{35}$$

which can be modeled using an extended formulation. A closed form expression for $\mathcal{T}_k$ in the original space remains elusive. However, one can algorithmically check if a given point (x^*, y^*) belongs to $\mathcal{T}_k$ by solving the separation problem. The following result is immediate using the polar description [79, Proposition I.4.5.1] of $\mathcal{T}_k$.

Proposition 2.6. *For any $j \in N, N_1 \subseteq I, N_2 \subseteq N \setminus j$, let $(\hat{x}_0^t(N_1, N_2), \hat{y}_j^t(N_1, N_2))$, for $t = 1, \ldots, k+1$, be the extreme points of $\mathcal{T}_k(N_1, N_2)$.*

Then, $(x^*, y^*) \in \mathcal{T}_k$ if and only if the optimal value of the following Cut Generating Linear Program (CGLP) is non-positive.

$$\begin{aligned}
\max \quad & \alpha^\top x^* + \beta^\top y^* - \gamma \\
s.t. \quad & \alpha_0 \, \hat{x}_0^t(N_1, N_2) + \beta_j \, \hat{y}_j^t(N_1, N_2) + \sum_{i \in N_1} \alpha_i + \sum_{i \in N_2} \beta_i u_i \quad \leq \gamma, \quad t = 1, \ldots, k+1, \\
& \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad j \in N, N_1 \subseteq I, N_2 \subseteq N \setminus j \\
& \alpha, \beta \in [-\mathbf{e}, \mathbf{e}], \; \gamma \in [-1, 1].
\end{aligned}$$

If the optimal value of this CGLP is positive and $(\alpha^*, \beta^*, \gamma^*)$ is an optimal solution, then ${\alpha^*}^\top x + {\beta^*}^\top y \leq \gamma^*$ is a valid inequality to $\mathcal{T}_k$, and hence $\text{conv}(\mathcal{P})$, that cuts off (x^*, y^*).

Note that the above CGLP has $2n+m+2$ variables but exponentially many constraints. The variable bound constraints in the CGLP are meant for normalization purposes.

In §2.3, we described standard polyhedral relaxations of $\text{conv}(\mathcal{P})$ using linearly many extra variables and constraints. However, since $\text{conv}(\mathcal{P})$, in general, is non-polyhedral, these relaxations may be weak. We now show $\mathcal{T}_k$ is a stronger relaxation than either of the two McCormick relaxations.

Proposition 2.7. *Consider a restriction $\mathcal{F}(N_1, N_2)$ of $\mathcal{P}$. Let $\mathcal{M}(N_1, N_2)$ and $\mathcal{SM}(N_1, N_2)$ be the projections of $\mathcal{M}(\mathcal{P})$ and $\mathcal{SM}(\mathcal{P})$, respectively, onto this subspace. Then,*

$$\mathcal{T}_2(N_1, N_2) \subset \mathcal{M}(N_1, N_2), \quad \mathcal{T}_2(N_1, N_2) \subset \mathcal{SM}(N_1, N_2).$$

Consequently, $\mathcal{T}_k \subset \text{Proj}_{x,y} \, \mathcal{M}(\mathcal{P})$ and $\mathcal{T}_k \subset \text{Proj}_{x,y} \, \mathcal{SM}(\mathcal{P})$ for any integer $k \geq 2$.

Proof. We only verify the inclusion of nontrivial right-leaning restrictions. The argument for left-leaning and trivial restrictions is similar. Let $\mathcal{F}(N_1, N_2) \in \mathfrak{F}^{\searrow}$.

Since $\bar{p}\lambda_j - \bar{q} > 0$, the domain $\mathcal{D} = [\ell, \vartheta] = [\bar{q}/\bar{p}, (\bar{q} + \lambda_j\mu_j)/(\bar{p} + \mu_j)]$. The two tangents for $\mathcal{T}_2(N_1, N_2)$ are drawn at the endpoints $(\ell, 0)$ and (ϑ, μ_j). First consider the restriction of $\mathcal{M}(\mathcal{P})$, denoted by $\mathcal{M}(N_1, N_2)$, onto the subspace $\{(x, y)\colon x_i = \bar{x}_i, i \in I, y_i = \bar{y}_i, i \in N \setminus j\}$ that defines this restriction $\mathcal{F}(N_1, N_2)$. Recall the definitions of $\bar{p}$ and $\bar{q}$ from (29). Since the McCormick envelopes (13) are exact at the bounds, it follows that

$$\bar{q} = \sum_{i \in N_1 \cap N_2} w_i + \sum_{i \in (N \setminus I) \cap N_2} a_{i-n} u_i. \tag{36}$$

Now $\sum_{i=1}^{n+m} y_i = y_j + \sum_{i \neq j} \bar{y}_i = y_j + \bar{p}$. After substituting this expression in the McCormick envelopes for w_0 in the definition of $\mathcal{M}(\mathcal{P})$ equation (25), we get

$$\begin{aligned}\mathcal{M}(N_1, N_2) = \big\{(x_0, y_j)\colon \exists w_0 \text{ s.t. } & \lambda_j y_j + \bar{q} = w_0,\ 0 \le y_j \le \mu_j \\ & [u_0 x_0 + y_j + \bar{p} - u_0]^+ \le w_0 \le \min\{u_0 x_0, y_j + \bar{p}\}\big\}.\end{aligned}$$

After projecting out w_0 using Fourier-Motzkin elimination, we get the polytope

$$\begin{aligned}\mathcal{M}(N_1, N_2) = \{(x_0, y_j)\colon\ & u_0 x_0 + (1 - \lambda_j) y_j \le \bar{q} - \bar{p} + u_0 \\ & -u_0 x_0 + \lambda_j y_j \le -\bar{q},\ 0 \le y_j \le \mu_j\}.\end{aligned}$$

It is easy to obtain the following four extreme points of $\mathcal{M}(N_1, N_2)$:

$$\begin{aligned}&\theta_1 = \left(\frac{\bar{q}}{u_0}, 0\right),\ \theta_2 = \left(\frac{u_0 + \bar{q} - \bar{p}}{u_0}, 0\right),\ \theta_3 = \left(\frac{\bar{q} + \lambda_j\mu_j}{u_0}, \mu_j\right), \\ &\theta_4 = \left(\frac{u_0 + \bar{q} - \bar{p} - (1 - \lambda_j)\mu_j}{u_0}, \mu_j\right),\end{aligned}$$

and check that they all lie outside $\mathcal{T}_2(N_1, N_2)$; see Figure 6(a). N_2 being a strictly independent set implies that $\bar{p} < u_0$ and hence θ_1 lies to the left to $(\ell, 0) = (\bar{q}/\bar{p}, 0)$. Now $\lambda_j \le 1$ and $\mu_j > 0$ imply $\bar{q} + \lambda_j\mu_j \le \bar{q} + \mu_j$. Hence $\vartheta = (\bar{q} + \lambda_j\mu_j)/(\bar{p} + \mu_j)$ lies to the left of $(\bar{q} + \mu_j)/(\bar{p} + \mu_j)$. Also, N_2 being independent gives us $\bar{p} + \mu_j < u_0$. Thus, θ_2 is to the right of $(\vartheta, 0)$. Since $\mathcal{M}(\mathcal{P})$ is a relaxation of $\text{conv}(\mathcal{P})$, it follows that $\mathcal{M}(N_1, N_2)$ is a relaxation of $\text{conv}(\mathcal{F}(N_1, N_2))$. Hence it must be that the points θ_3 and θ_4 are to the left and right of (ϑ, μ_j), respectively.

Now consider $\mathcal{SM}(N_1, N_2)$. Again, the exactness of McCormick envelopes at the bounds gives us (36) and $\sum_{i \neq j} w_{0i} = \bar{p} x_0$. Hence, this restriction of $\mathcal{SM}(\mathcal{P})$ is

$$\begin{aligned}\mathcal{SM}(N_1, N_2) = \big\{(x_0, y_j) \colon \exists w_{0j} \text{ s.t. } & \lambda_j y_j + \bar{q} = w_{0j} + \bar{p} x_0,\ 0 \leq y_j \leq \mu_j \\ & [u_j x_0 + y_j - u_j]^+ \leq w_{0j} \leq \min\{u_j x_0, y_j\}\big\}.\end{aligned}$$

Upon projecting out w_{0j}, we get

$$\begin{aligned}\mathcal{SM}(N_1, N_2) = \big\{(x_0, y_j) \colon & \bar{p} x_0 - \lambda_j y_j \leq \bar{q},\ 0 \leq y_j \leq \mu_j \\ & (\bar{p} + u_j) x_0 + (1 - \lambda_j) y_j \leq \bar{q} + u_j \\ & -(\bar{p} + u_j) x_0 + \lambda_j y_j \leq -\bar{q} \\ & -\bar{p} x_0 - (1 - \lambda_j) y_j \leq -\bar{q}\big\}.\end{aligned}$$

The extreme points of $\mathcal{SM}(N_1, N_2)$ are (cf. Figure 6(b))

$$\rho_1 = \left(\frac{\bar{q}}{\bar{p}}, 0\right),\ \rho_2 = \left(\frac{\bar{q} + \lambda_j u_j}{\bar{p} + \lambda_j u_j}, \frac{(\bar{p} - \bar{q}) u_j}{\bar{p} + \lambda_j u_j}\right),\ \rho_3 = \left(\frac{\bar{q}}{\bar{p} + (1 - \lambda_j) u_j}, \frac{u_j \bar{q}}{\bar{p} + (1 - \lambda_j) u_j}\right),$$
$$\rho_4 = \left(\frac{\bar{q} + \lambda_j \mu_j}{\bar{p} + u_j}, \mu_j\right),\ \rho_5 = \left(\frac{\bar{q} + u_j - (1 - \lambda_j)\mu_j}{\bar{p} + u_j}, \mu_j\right).$$

The first extreme point ρ_1 coincides with $(\ell, 0)$. The third point ρ_3 is to the left of ρ_1 because $\lambda_j \leq 1$. The fourth point ρ_4 is to the left of (ϑ, μ_j) because $u_j \geq \mu_j$. The fifth point ρ_5 is to the right of (ϑ, μ_j) because

$$\begin{aligned} & \frac{\bar{q} + u_j - (1 - \lambda_j)\mu_j}{\bar{p} + u_j} \geq \frac{\bar{q} + \lambda_j \mu_j}{\bar{p} + \mu_j} \\ \iff \quad & (\bar{q} + \lambda_j \mu_j)\left[\frac{1}{\bar{p} + u_j} - \frac{1}{\bar{p} + \mu_j}\right] \geq \frac{\mu_j - u_j}{\bar{p} + u_j} \\ \iff \quad & \frac{\bar{q} + \lambda_j \mu_j}{\bar{p} + \mu_j} \leq 1\end{aligned}$$

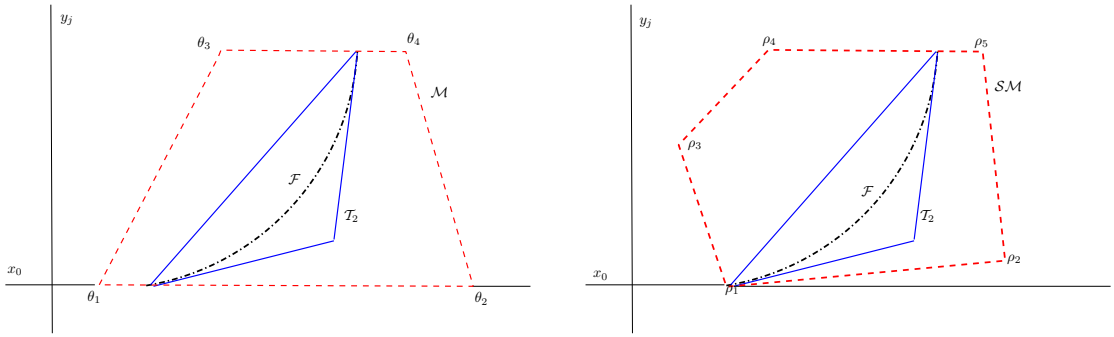
which is true since $\bar{q} \leq \bar{p}$ and $\lambda_j \leq 1$. Now, ρ_2 and ρ_5 lie on $(\bar{p} + u_j) x_0 + (1 - \lambda_j) y_j \leq \bar{q} + u_j$, which has a positive slope, and ρ_1 and ρ_2 lie on $\bar{p} x_0 - \lambda_j y_j \leq \bar{q}$, which makes a smaller slope than the tangent at ρ_1 and the segment joining ρ_1 and ρ_5, due to $\mathcal{SM}(N_1, N_2)$ being a relaxation of $\mathcal{F}(N_1, N_2)$. Hence, ρ_2 must be to the right of ρ_5.

Under the assumption that the two endpoints, $(\ell, 0)$ and (ϑ, μ_j), are always selected for gradient generation in (34), it is clear that $\mathcal{T}_k(N_1, N_2) \subset \mathcal{T}_2(N_1, N_2)$ for all $k \geq 3$. Thus,

$\mathcal{T}_k(N_1, N_2) \subset \mathcal{M}(N_1, N_2)$ and $\mathcal{T}_k(N_1, N_2) \subset \mathcal{SM}(N_1, N_2)$, which implies

$$\begin{aligned} \mathcal{T}_k &= \text{conv} \bigcup_{j \in N} \bigcup_{\substack{N_1 \subseteq I \\ N_2 \subseteq \bar{N} \setminus j}} \mathcal{T}_k(N_1, N_2) \\ &\subset \text{conv} \bigcup_{j \in N} \bigcup_{\substack{N_1 \subseteq I \\ N_2 \subseteq \bar{N} \setminus j}} \mathcal{M}(N_1, N_2) \\ &\subseteq \text{Proj}_{x,y} \mathcal{M}(\mathcal{P}), \end{aligned}$$

where the last inclusion is due to $\mathcal{M}(N_1, N_2) \subset \text{Proj}_{x,y} \mathcal{M}(\mathcal{P})$ for all j, N_1, N_2, and commutivity of projection and convex hull operators. Similarly for $\mathcal{T}_k \subset \text{Proj}_{x,y} \mathcal{SM}(\mathcal{P})$. □



(a) McCormick $\mathcal{M}(N_1, N_2)$ (b) McCormick $\mathcal{SM}(N_1, N_2)$

Figure 6: Comparing $\mathcal{F}(N_1, N_2)$ and $\mathcal{T}_2(N_1, N_2)$ with restrictions of McCormick relaxations $\mathcal{M}(\mathcal{P})$ and $\mathcal{SM}(\mathcal{P})$.

2.5 Lifted inequalities

Proposition 2.6 provides an implicit way of generating valid inequalities to conv($\mathcal{P}$). However, it requires solving a CGLP with exponentially many constraints (one for each restriction of $\mathcal{P}$), which can prove intractable for problems of reasonable size. Here we derive tools to obtain explicit valid inequalities to conv($\mathcal{P}$).

Consider a restriction $\mathcal{F}(N_1, N_2)$ and let $\alpha_0 x_0 + \beta_j y_j \leq \gamma$ be valid to it. We refer to this inequality as a *seed inequality* for lifting. Some obvious choices for this seed inequality include facets of $\mathcal{T}_k(N_1, N_2)$ or the facets of the projected McCormick relaxations $\mathcal{M}(N_1, N_2)$ and $\mathcal{SM}(N_1, N_2)$ (note that the latter are weaker than former due to Proposition 2.7). Our

objective is to find coefficients $\alpha \in \Re^{n+1}$ and $\beta \in \Re^{n+m}$ such that

$$\sum_{i=1}^{n} \alpha_i(x_i - \bar{x}_i) + \sum_{i=1}^{n+m} \beta_i(y_i - \bar{y}_i) \leq \gamma \tag{37}$$

is a valid inequality to the convex hull of $\mathcal{P}$. Here, $(\bar{x}, \bar{y})$ is a fixed value of (x, y). Thus, $\bar{x}_i = 1, \forall i \in N_1, \bar{x}_i = 0, \forall i \in N_1^-, y_i = u_i, \forall i \in N_2, y_i = 0, \forall i \in N_2^-$ and for convenience, we denote $\bar{x}_0 = \bar{y}_j = 0$.

As a first step in this procedure, we find α_j by lifting the variable x_j, when $j \in I$. To do this, consider the following lemma that provides the maximum value attained by any linear function over $\mathcal{F}(N_1, N_2)$.

Lemma 2.4. *Consider optimizing a linear function over $\mathcal{F}(N_1, N_2)$.*

$$\nu^*(c, d, N_1^-, N_1) = \max\{cx_0 + dy_j \colon (x_0, y_j) \in \mathcal{F}(N_1, N_2)\}.$$

Let $\ell = \bar{q}/\bar{p}$ and $\vartheta = (\bar{q} + \lambda_j\mu_j)/(\bar{p} + \mu_j)$ for nontrivial $\mathcal{F}(N_1, N_2)$. The optimal value ν^* is given as follows.

$\mathcal{F}(N_1, N_2) \in \mathfrak{F}^{\searrow}$ **:**

$$\nu^*(c, d, N_1^-, N_1) = \begin{cases} c\ell, & c < 0, d < 0 \\ c\vartheta + d\mu_j, & c \geq 0, d \geq 0 \\ \max\{c\ell, c\vartheta + d\mu_j\}, & c \leq 0, d \geq 0 \\ c\lambda_j - d\bar{p} - 2\sqrt{cd(\bar{q} - \bar{p}\lambda_j)}, & c > 0, d < 0, \dfrac{\bar{p}^2}{\bar{p}\lambda_j - \bar{q}} \leq \dfrac{-c}{d} \leq \dfrac{(\bar{p} + \mu_j)^2}{\bar{p}\lambda_j - \bar{q}} \\ c\ell, & c > 0, d < 0, \dfrac{-c}{d} < \dfrac{\bar{p}^2}{\bar{p}\lambda_j - \bar{q}} \\ c\vartheta + d\mu_j, & c > 0, d < 0, \dfrac{-c}{d} > \dfrac{(\bar{p} + \mu_j)^2}{\bar{p}\lambda_j - \bar{q}}. \end{cases}$$

$\mathcal{F}(N_1, N_2) \in \mathfrak{F}^{\swarrow}$ **:**

$$\nu^*(c, d, N_1^-, N_1) = \begin{cases} c\lambda_j - d\bar{p} - 2\sqrt{cd(\bar{q} - \bar{p}\lambda_j)}, & c < 0, d < 0, \dfrac{\bar{p}^2}{\bar{q} - \lambda_j\bar{p}} \leq \dfrac{c}{d} \leq \dfrac{(\bar{p} + \mu_j)^2}{\bar{q} - \lambda_j\bar{p}} \\ c\ell, & c < 0, d < 0, \dfrac{c}{d} < \dfrac{\bar{p}^2}{\bar{q} - \lambda_j\bar{p}} \\ c\vartheta + d\mu_j, & c < 0, d < 0, \dfrac{c}{d} > \dfrac{(\bar{p} + \mu_j)^2}{\bar{q} - \lambda_j\bar{p}} \\ \max\{c\ell, c\vartheta + d\mu_j\}, & c \geq 0, d \geq 0 \\ c\vartheta + d\mu_j, & c \leq 0, d \geq 0 \\ c\ell, & c > 0, d < 0. \end{cases}$$

$\mathcal{F}(N_1, N_2)$ **is trivial with** $\bar{q} = \bar{p} = 0$ **:** $\nu^*(c, d, N_1^-, N_1) = \max\{0,\, c,\, c\lambda_j + d\mu_j\}$.

$\mathcal{F}(N_1, N_2)$ **is trivial with** $\bar{p} > 0, \bar{q} = \lambda_j\bar{p}$ **:** $\nu^*(c, d, N_1^-, N_1) = c\lambda_j + \max\{0, d\}\mu_j$.

Proof. Since the objective function is linear in (x_0, y_j) and $\mathrm{conv}(\mathcal{F}(N_1, N_2))$ is compact, the optimum always exists and is attained at some extreme point of $\mathrm{conv}(\mathcal{F}(N_1, N_2))$. We first resolve the trivial cases. If $\bar{q} = \bar{p} = 0$, then $\mathrm{conv}(\mathcal{F}(N_1, N_2))$ is a simplex formed by $\{(0, 0), (0, 1), (\lambda_j, \mu_j)\}$. Else if $\bar{q} = \lambda_j\bar{p}$, then $\mathcal{F}(N_1, N_2)$ is simply a segment between $(\lambda_j, 0)$ and (λ_j, μ_j).

Now let us assume that the restriction is nontrivial. We only address the right-leaning case, since the left-leaning case is analogous, subject to minor sign reversals. Hence $\mathcal{F}(N_1, N_2) \in \mathfrak{F}^{\searrow}$. Most of the cases follow from the graphical illustration of $\mathcal{F}(N_1, N_2)$ in Figure 5(a). For $c < 0, d = 0$ or $c = 0, d < 0$, it is obvious that the maximum is attained at $(\ell, 0)$. Since $\psi_j(\cdot)$ is nondecreasing on $[0, \lambda_j)$, then for any $c < 0, d < 0$, the maximum is again at $(\ell, 0)$. For $c = 0, d \geq 0$ or $c \geq 0, d = 0$, the maximum is at (ϑ, μ_j), and hence also for any $c, d \geq 0$. For $c \leq 0, d \geq 0$, the maximum is at one of the two extreme points $(\ell, 0)$ or (ϑ, μ_j). The nontrivial case to check is when $c > 0, d < 0$. In this case, the optimum solution may be attained at a point $(\hat{x}_0, \hat{y}_j)$ on $y_j = \psi_j(x_0)$ such that (c, d) is linearly dependent on the gradient at this point. Formally, for some $\tau \neq 0$,

$$c = \tau\,\frac{\bar{p}\lambda_j - \bar{q}}{(\lambda_j - \hat{x}_0)^2}, \qquad d = -\tau.$$

After rearranging terms,

$$\hat{x}_0 = \lambda_j - \sqrt{\frac{(-d)(\bar{p}\lambda_j - \bar{q})}{c}} \quad \Rightarrow \quad \hat{y}_j = -\bar{p} + \sqrt{\frac{c(\bar{p}\lambda_j - \bar{q})}{-d}}.$$

For this point to be optimal, we must have $\hat{x}_0 \in \mathcal{D}$ or equivalently $\hat{y}_j \in [0, \mu_j]$. Hence we derive that

$$\frac{\bar{p}^2}{\bar{p}\lambda_j - \bar{q}} \leq \frac{-c}{d} \leq \frac{(\bar{p} + \mu_j)^2}{\bar{p}\lambda_j - \bar{q}} \tag{38}$$

As long as (38) is satisfied, the optimal value is

$$\begin{aligned} c\hat{x}_0 + d\hat{y}_j &= c\lambda_j - \sqrt{c(-d)(\bar{p}\lambda_j - \bar{q})} - d\bar{p} - \sqrt{-d}\sqrt{c(\bar{p}\lambda_j - \bar{q})} \\ &= c\lambda_j - d\bar{p} - 2\sqrt{cd(\bar{q} - \bar{p}\lambda_j)}. \end{aligned}$$

If (38) is violated, then the optimal lies at either $(\ell, 0)$ or (ϑ, μ_j), depending on the violated bound. □

Now suppose that $\alpha_0 x_0 + \beta_j y_j \leq \gamma$ is a valid inequality to $\mathcal{F}(N_1, N_2)$. Assume, for simplicity, that $j \in N_1$. Then to lift x_j, we must find α_j such that

$$\alpha_0 x_0 + \beta_j y_j + \alpha_j(\phi_1 - 1) \leq \gamma, \quad \forall \phi_1 \in [0, 1]. \tag{39}$$

Note that since $\text{conv}(\mathcal{P})$ is the convex hull of its extreme points, it suffices to make the seed inequality valid to every extreme point of $\text{conv}(\mathcal{P})$. Theorem 2.1 dictates that $x_j \in \{0, 1\}$ at every extreme point of $\text{conv}(\mathcal{P})$, and hence we are allowed to replace $\phi_1 \in [0, 1]$ in (39) by $\phi_1 \in \{0, 1\}$. Since the inequality is already valid for $\phi_1 = 1$ (by assumption $j \in N_1$), it must be that

$$\begin{aligned} \alpha_j &\geq -\gamma + \max\{\alpha_0 x_0 + \beta_j y_j \colon (x_0, y_j) \in \mathcal{F}(N_1^- \cup j, N_1 \setminus j, N_2^-, N_2)\} \\ &= -\gamma + \nu^*(\alpha_0, \beta_j, N_1^- \cup j, N_1 \setminus j). \end{aligned}$$

Now the value of ν^* can be computed using Lemma 2.4, which leads us to the next result.

Proposition 2.8. *Let $j \in I$.*

1. *Suppose that $\mathcal{F}(N_1, N_2)$ is nontrivial and $\alpha_0 x_0 + \beta_j y_j \leq \gamma$ is a non-secant facet of $\mathcal{T}_k(N_1, N_2)$, for some $k \geq 2$. For $j \in N_1$, the lifted secant*

$$-\mu_j x_0 + \frac{(\bar{p} - \bar{q})\mu_j}{\bar{p}(\bar{p} + \mu_j)} y_j + \frac{\bar{p}\mu_j}{\bar{p} + \mu_j} x_j \leq \frac{\bar{p}\mu_j}{\bar{p} + \mu_j} - \frac{\bar{q}\mu_j}{\bar{p}}$$

and the lifted gradient

$$\alpha_0 x_0 + \beta_j y_j \; - \; \left(\gamma - \nu^*(\alpha_0, \beta_j, N_1^- \cup j, N_1 \setminus j)\right) x_j \leq \nu^*(\alpha_0, \beta_j, N_1^- \cup j, N_1 \setminus j)$$

are valid to $\mathcal{F}(N_1^-, N_1 \setminus j, N_2^-, N_2)$. *Similarly, for* $j \in N_1^-$,

$$\begin{aligned} \mu_j x_0 + \frac{\bar{q}\mu_j}{\bar{p}(\bar{p}+\mu_j)} y_j - \frac{\mu_j^2}{\bar{p}+\mu_j} x_j &\leq \frac{\bar{q}\mu_j}{\bar{p}} \\ \alpha_0 x_0 + \beta_j y_j \; + \; \left(\gamma - \nu^*(\alpha_0, \beta_j, N_1^- \setminus j, N_1 \cup j)\right) x_j &\leq \gamma \end{aligned}$$

are valid to $\mathcal{F}(N_1^- \setminus j, N_1, N_2^-, N_2)$.

2. *If* $\mathcal{F}(N_1, N_2) = \mathcal{F}(N_1, \emptyset)$ *is trivial with* $\bar{p} = \bar{q} = 0$, *then the secants*

$$\begin{aligned} -u_j x_0 + y_j + u_j x_j &\leq u_j \qquad j \in N_1 \\ u_j x_0 + y_j - u_j x_j &\leq u_j \qquad j \in N_1^- \end{aligned}$$

are valid to $\mathcal{F}(N_1^-, N_1 \setminus j, I \setminus j, \emptyset)$ *and* $\mathcal{F}(N_1^- \setminus j, N_1, I \setminus j, \emptyset)$, *respectively.*

Thus, at this stage we have derived inequalities in the (x_0, y_j, x_j)-space (if $j \notin I$, then there is no x_j and we simply ignore the above lifting step). We now lift the remaining variables. We derive a valid inequality to $\mathcal{P}$ of the type (46) by lifting pairs of variables (x_i, y_i), for all $i \in N \setminus j$, into a inequality from Proposition 2.8. Henceforth, if $i \notin I$, then the pair (x_i, y_i) is to be interpreted as (a_{i-n}, y_i), i.e. we are effectively lifting only the variable y_i.

2.5.1 Pairwise sequence independent lifting

Suppose that after l stages of the lifting procedure, we have lifted pairs of variables in the subset $L \subseteq N$. Let $y \in L$ and $|L| = l \geq 1$. Denote this set, in which all the variables from $N \setminus (L \cup \{0\})$ are fixed, as $\mathcal{P}(N \setminus L)$. Thus, we have a inequality

$$\alpha_0 x_0 + \sum_{i \in L \cap I} \alpha_i (x_i - \bar{x}_i) + \beta_j y_j + \sum_{i \in L \setminus j} \beta_i (y_i - \bar{y}_i) \; \leq \; \gamma, \tag{40}$$

that is valid to the convex hull of $\mathcal{P}(N \setminus L)$. Let us define a perturbation function with respect to (40) as follows.

$$\begin{aligned}
\Upsilon_l(\delta,\Delta) := \quad \gamma \; - \max \quad & \alpha_0 x_0 + \sum_{i\in L\cap I} \alpha_i(x_i - \bar{x}_i) + \beta_j y_j + \sum_{i\in L\setminus j} \beta_i(y_i - \bar{y}_i) \\
\text{s.t.} \quad & \sum_{i\in L\cap I} x_i y_i + \sum_{i\in L\setminus I} a_{i-n} y_i + \sum_{i\in I\setminus L} \bar{x}_i \bar{y}_i + \sum_{i\in N\setminus(I\cup L)} a_{i-n}\bar{y}_i + \delta \\
& \qquad = x_0 \left(\sum_{i\in L} y_i + \sum_{i\in N\setminus L} \bar{y}_i + \Delta \right) \\
& \sum_{i\in L} y_i + \sum_{i\in N\setminus L} \bar{y}_i + \Delta \;\le\; u_0 \\
& x_0 \in [0,1],\; x_i \in \{0,1\}, y_i \in [0,u_i],\; i \in L.
\end{aligned} \tag{41}$$

$\Upsilon^l \colon \Re^2 \mapsto \Re$ where δ represents the perturbation in the bilinear term on the left hand side $\sum_{i\in I\setminus L} x_i y_i + \sum_{i\in N\setminus(I\cup L)} a_{i-n} y_i$, and Δ is the perturbation in $\sum_{i\in N\setminus L} y_i$. This is necessary in order to capture the presence of y_i on both the left and right hand sides of the bilinear equality constraint. Observe that similar to the lifting step of x_j, we have replaced $x_i \in [0,1]$ with $x_i \in \{0,1\}$ in (41), based on the extreme point characterization of Theorem 2.1.

We first show that if $\Upsilon_l(\delta,\Delta)$ can be underestimated by a linear function, then it yields valid coefficients for a new pair $(x_{i_{l+1}}, y_{i_{l+1}})$.

Proposition 2.9. *Consider a valid inequality* (40) *to* $\mathcal{P}(N \setminus L)$*, for some* $l \ge 1$*. For* $i_{l+1} \in I \setminus L$*, assume that there exist reals* $\alpha_{i_{l+1}}$ *and* $\beta_{i_{l+1}}$ *such that*

$$\alpha_{i_{l+1}}(\phi_1 - \bar{x}_{i_{l+t}}) + \beta_{i_{l+1}}(\phi_2 - \bar{y}_{i_{l+t}}) \le \Upsilon_l(\phi_1\phi_2 - \bar{x}_{i_{l+1}}\bar{y}_{i_{l+1}}, \phi_2 - \bar{y}_{i_{l+1}}),\; \forall \phi_1 \in \{0,1\}, \phi_2 \in [0, u_{i_{l+1}}]. \tag{42}$$

Then the inequality

$$\alpha_0 x_0 + \sum_{i\in L\cap I} \alpha_i(x_i - \bar{x}_i) + \alpha_{i_{l+1}}(x_{i_{l+1}} - \bar{x}_{i_{l+1}}) + \beta_j y_j + \sum_{i\in L\setminus j} \beta_i(y_i - \bar{y}_i) + \beta_{i_{l+1}}(y_{i_{l+1}} - \bar{y}_{i_{l+1}}) \;\le\; \gamma,$$

is valid to $\mathcal{P}(N \setminus (L \cup i_{l+1}))$*.*

Similarly, for $i_{l+1} \in N \setminus (I \cup L)$*, if there exists a* $\beta_{i_{l+1}}$ *such that*

$$\beta_{i_{l+1}}(\phi_2 - \bar{y}_{i_{l+t}}) \le \Upsilon_l(a_{i_{l+1}-n}(\phi_2 - \bar{y}_{i_{l+1}}), \phi_2 - \bar{y}_{i_{l+1}}), \quad \forall \phi_2 \in [0, u_{i_{l+1}}],$$

then the lifted valid inequality to $\mathcal{P}(N \setminus (L \cup i_{l+1}))$ *is*

$$\alpha_0 x_0 + \sum_{i \in L \cap I} \alpha_i (x_i - \bar{x}_i) + \beta_j y_j + \sum_{i \in L \setminus j} \beta_i (y_i - \bar{y}_i) + \beta_{i_{l+1}} (y_{i_{l+1}} - \bar{y}_{i_{l+1}}) \ \leq \ \gamma.$$

Proof. First consider $i_{l+1} \in I$. By definition (41) of the perturbation function, we know that for any $\phi_1 \in \{0,1\}, \phi_2 \in [0, u_{i_{l+1}}]$ and (x, y) that satisfies capacity constraint and the bilinear equality in (41) with $\delta = \phi_1 \phi_2 - \bar{x}_{i_{l+1}} \bar{y}_{i_{l+1}}$ and $\Delta = \phi_2 - \bar{y}_{i_{l+1}}$,

$$\Upsilon_l(\phi_1 \phi_2 - \bar{x}_{i_{l+1}} \bar{y}_{i_{l+1}}, \phi_2 - \bar{y}_{i_{l+1}}) \leq \gamma - \alpha_0 x_0 + \sum_{i \in L \cap I} \alpha_i (x_i - \bar{x}_i) + \beta_j y_j + \sum_{i \in L \setminus j} \beta_i (y_i - \bar{y}_i).$$

If there exist $\alpha_{i_{l+1}}$ and $\beta_{i_{l+1}}$ such that (42) is true, then it follows that the proposed inequality is valid to $\mathcal{P}(N \setminus (L \cup i_{l+1}))$. The proof for $j \in N \setminus I$ follows similarly. □

Starting with the inequality (40) that is valid to $\text{conv}(\mathcal{P}(N \setminus L))$, a repeated application of Proposition 2.9 produces a valid inequality to $\text{conv}(\mathcal{P})$. However, this type of sequential lifting has a drawback in that it requires the computation of the perturbation function $\Upsilon_{l+t}(\delta, \Delta)$ at each step $t \geq 1$. Note that (41) requires maximization over a bilinear equality set, similar in structure to our original set $\mathcal{P}$, and hence we expect the computation of $\Upsilon_{l+t}(\delta, \Delta)$ to be a difficult problem. To overcome this issue, we would ideally like to perform a single step by obtaining lifting coefficients for all the remaining variables with respect to $\Upsilon_l(\delta, \Delta)$. This is guaranteed to happen if we can show that $\Upsilon_l(\delta, \Delta) = \Upsilon_{l+t}(\delta, \Delta)$ for all $t \geq 1$. When this is true, we say that lifting is *sequence independent*. The next result gives a sufficient condition, akin to integer programming, for sequence independent lifting. Before presenting this sufficient condition, we comment on the domain of $\Upsilon_l(\cdot, \cdot)$. Since δ and Δ represent perturbations in $\sum_{i \in I \setminus L} x_i y_i + \sum_{i \in N \setminus (I \cup L)} a_{i-n} y_i$ and $\sum_{i \in N \setminus L} y_i$, respectively, and since $x_i, a_{i-n} \in [0,1]$ and $y_i \in [0, u_i]$, it follows that the values of interest for (δ, Δ) lie in a set $\mathcal{R}_l$ defined as

$$\begin{aligned} \mathcal{R}_l = \Big\{ (\delta, \Delta) \in \Re^2 \colon 0 \leq \sum_{i \in N \setminus L} \bar{y}_i + \Delta \leq U_l \\ 0 \leq \sum_{i \in I \setminus L} \bar{x}_i \bar{y}_i + \sum_{i \in N \setminus (I \cup L)} a_{i-n} y_i + \delta \leq \sum_{i \in N \setminus L} \bar{y}_i + \Delta \Big\}, \end{aligned} \tag{43}$$

where $U_l = \min\{u_0, \sum_{k\in N\setminus L} u_k\}$. We say that $\Upsilon_l(\cdot,\cdot)$ is *jointly superadditive* over $\mathcal{R}_l$ if for any $(\delta^1,\Delta^1),(\delta^2,\Delta^2) \in \mathcal{R}_l$ such that $(\delta^1+\delta^2,\Delta^1+\Delta^2) \in \mathcal{R}_l$, we have

$$\Upsilon_l(\delta^1+\delta^2,\Delta^1+\Delta^2) \geq \Upsilon_l(\delta^1,\Delta^1) + \Upsilon_l(\delta^2,\Delta^2).$$

Proposition 2.10. *Consider a valid inequality* (40) *to* $\mathcal{P}(N\setminus L)$*, for some* $l \geq 1$*. Assume that for every* $t \geq 1$ *and*

1. *for* $i_{l+t} \in I \setminus L$*, there exist reals* $\alpha_{i_{l+t}}$ *and* $\beta_{i_{l+t}}$ *such that for all* $\phi_1 \in \{0,1\}$ *and* $\phi_2 \in [0, u_{i_{l+t}}]$*,*

$$\alpha_{i_{l+t}}(\phi_1 - \bar{x}_{i_{l+t}}) + \beta_{i_{l+t}}(\phi_2 - \bar{y}_{i_{l+t}}) \leq \Upsilon_l(\phi_1\phi_2 - \bar{x}_{i_{l+t}}\bar{y}_{i_{l+t}}, \phi_2 - \bar{y}_{i_{l+t}}). \tag{44}$$

2. *for* $i_{l+t} \in N \setminus (I \cup L)$*, there exists a real* $\beta_{i_{l+t}}$ *such that for all* $\phi_2 \in [0, u_{i_{l+t}}]$*,*

$$\beta_{i_{l+t}}(\phi_2 - \bar{y}_{i_{l+t}}) \leq \Upsilon_l(a_{i_{l+t}-n}(\phi_2 - \bar{y}_{i_{l+t}}), \phi_2 - \bar{y}_{i_{l+t}}). \tag{45}$$

If $\Upsilon_l(\cdot,\cdot)$ *is jointly superadditive over* $\mathcal{R}_l$*, then*

$$\alpha_0 x_0 + \sum_{i\in I} \alpha_i(x_i - \bar{x}_i) + \beta_j y_j + \sum_{i\in N\setminus j} \beta_i(y_i - \bar{y}_i) \;\leq\; \gamma \tag{46}$$

is a valid inequality to $\mathrm{conv}(\mathcal{P})$*.*

Proof. Under the given assumptions and the result of Proposition 2.9, it suffices to prove our claim that $\Upsilon_{l+t}(\delta,\Delta) = \Upsilon_l(\delta,\Delta)$ for all $t \geq 0$ and $(\delta,\Delta) \in \mathcal{R}_l$. First note that $\Upsilon_{l+t}(\cdot,\cdot) \leq \Upsilon_{l+t-1}(\cdot,\cdot)$. This is because for the function $\Upsilon_{l+t}(\delta,\Delta)$, one extra pair of variable has been lifted compared to $\Upsilon_{l+t-1}(\delta,\Delta)$. Hence the maximization while computing $\Upsilon_{l+t-1}(\delta,\Delta)$ is taken over a restriction of the feasible set in computing $\Upsilon_{l+t}(\delta,\Delta)$. Also, the extra linear term $\alpha_{i_{l+t}}(x_{i_{l+t}} - \bar{x}_{i_{l+t}}) + \beta_{i_{l+t}}(y_{i_{l+t}} - \bar{y}_{i_{l+t}})$ in the objective of $\Upsilon_{l+t}(\cdot,\cdot)$ vanishes to zero when $(x_{i_{l+t}}, y_{i_{l+t}})$ is fixed to $(\bar{x}_{i_{l+t}}, \bar{y}_{i_{l+t}})$. This implies $\Upsilon_{l+t}(\cdot,\cdot) \leq \Upsilon_{l+t-1}(\cdot,\cdot)$. Since this is true for all t, it follows that $\Upsilon_{l+t}(\delta,\Delta) \leq \Upsilon_l(\delta,\Delta)$.

The reverse inequality $\Upsilon_{l+t}(\cdot,\cdot) \geq \Upsilon_l(\cdot,\cdot)$ is proven by induction on t. The base case $t = 0$ is trivially true. As part of induction hypothesis, it is assumed to be true for $\Upsilon_{l+t-1}(\cdot,\cdot)$. First, let us consider $i_{l+t} \in I$. In the maximization problem for $\Upsilon_{l+t}(\cdot,\cdot)$,

suppose that we fix $x_{i_{l+t}} = \phi_1$ and $y_{i_{l+t}} = \phi_2$ for some $\phi_1 \in \{0,1\}$ and $\phi_2 \in [0, u_{i_{l+t}}]$. This provides the following relationship between $\Upsilon_{l+t-1}(\cdot,\cdot)$ and $\Upsilon_{l+t}(\cdot,\cdot)$

$$\begin{aligned}\Upsilon_{l+t}(\delta,\Delta) \;=\; \inf \quad & \Upsilon_{l+t-1}(\delta + \phi_1\phi_2 - \bar{x}_{i_{l+t}}\bar{y}_{i_{l+t}}, \Delta + \phi_2 - \bar{y}_{i_{l+t}}) \\ & - \;\alpha_{i_{l+t}}(\phi_1 - \bar{x}_{i_{l+t}}) \;-\; \beta_{i_{l+t}}(\phi_2 - \bar{y}_{i_{l+t}}) \qquad (47)\\ \text{s.t.} \quad & \phi_1 \in \{0,1\}, \phi_2 \in [0, u_{i_{l+t}}].\end{aligned}$$

By induction hypothesis, we can replace $\Upsilon_{l+t-1}(\cdot,\cdot)$ with $\Upsilon_l(\cdot,\cdot)$ in above equality. Observe that $(\phi_1\phi_2 - \bar{x}_{i_{l+t}}\bar{y}_{i_{l+t}}, \phi_2 - \bar{y}_{i_{l+t}}) \in \mathcal{R}_l$. Joint superadditivity of $\Upsilon_l(\cdot,\cdot)$ then implies that

$$\begin{aligned}\Upsilon_{l+t}(\delta,\Delta) \;\geq\; \Upsilon_l(\delta,\Delta) \;+\; \inf \quad & \Upsilon_l(\phi_1\phi_2 - \bar{x}_{i_{l+t}}\bar{y}_{i_{l+t}}, \phi_2 - \bar{y}_{i_{l+t}}) \\ & - \;\alpha_{i_{l+t}}(\phi_1 - \bar{x}_{i_{l+t}}) \;-\; \beta_{i_{l+t}}(\phi_2 - \bar{y}_{i_{l+t}}) \\ \text{s.t.} \quad & \phi_1 \in \{0,1\}, \phi_2 \in [0, u_{i_{l+t}}].\end{aligned}$$

The infimum in the above inequality is nonnegative due to the assumption of (44). Hence, $\Upsilon_{l+t}(\cdot,\cdot) \geq \Upsilon_l(\cdot,\cdot)$ and the induction step is complete.

For $i_{l+t} \in N \setminus I$, we only fix $y_{i_{l+t}} = \phi_2$ in the maximization problem for $\Upsilon_{l+t}(\delta,\Delta)$. Hence,

$$\Upsilon_{l+t}(\delta,\Delta) = \inf_{\phi_2 \in [0, u_{i_{l+t}}]} \left\{ \Upsilon_{l+t-1}(\delta + a_{i_{l+t}-n}(\phi_2 - \bar{y}_{i_{l+t}}), \Delta + \phi_2 - \bar{y}_{i_{l+t}}) - \beta_{i_{l+t}}(\phi_2 - \bar{y}_{i_{l+t}}) \right\},$$

and the remaining steps are similar. □

The above proposition greatly relies on superadditivity of $\Upsilon_l(\cdot,\cdot)$. In fact, it is crucial in obtaining the relation $\Upsilon_{l+t}(\cdot,\cdot) \geq \Upsilon_l(\cdot,\cdot)$. Unfortunately, many perturbation functions are not jointly superadditive. In order to perform sequence independent lifting in this case, we extend the result of Gu et al. [53], Theorem 3, to our context. As in integer programming, we call a function $\varphi_l(\cdot,\cdot)$ to be a *valid lifting function* if it is jointly superadditive and underestimates the true lifting function $\Upsilon_l(\cdot,\cdot)$.

Proposition 2.11. *Let $\varphi_l(\cdot,\cdot)$ be a valid lifting function and suppose that the coefficients $\alpha_{i_{l+t}}$ and $\beta_{i_{l+t}}$ in* (44) *and* (45) *are computed with respect to $\varphi_l(\cdot,\cdot)$. Then* (46) *is a valid inequality to* $\text{conv}(\mathcal{P})$.

Proof. The proof follows similar steps as Gu et al. [53], Theorem 3. Let $\varphi_l(\delta, \Delta) \leq \Upsilon_l(\delta, \Delta), \forall (\delta, \Delta) \in \mathcal{R}_l$, be jointly superadditive. We will prove by induction that $\varphi_l(\cdot, \cdot) \leq \Upsilon_{l+t}(\cdot, \cdot)$ for all t. The base case $t = 0$ is obvious by validity of $\varphi_l(\cdot, \cdot)$. Now assume that $\varphi_l(\cdot, \cdot) \leq \Upsilon_{l+t-1}(\cdot, \cdot)$ for some $t \geq 1$. Let (x^*, y^*) be an optimal solution to the maximization problem in $\Upsilon_{l+t}(\delta, \Delta)$. Thus,

$$\Upsilon_{l+t}(\delta, \Delta) = \gamma - \Bigg[\alpha_0 x_0^* + \sum_{i \in L \cap I} \alpha_i (x_i^* - \bar{x}_i) + \sum_{i=i_{l+1}}^{i_{l+t}} \alpha_i (x_i^* - \bar{x}_i) + \beta_j y_j^* + \sum_{i \in L \setminus j} \beta_i (y_i^* - \bar{y}_i) + \sum_{i=i_{l+1}}^{i_{l+t}} \beta_i (y_i^* - \bar{y}_i)\Bigg].$$

From (47), we have

$$\begin{aligned}\Upsilon_{l+t}(\delta, \Delta) =& \Upsilon_{l+t-1}(\delta + x^*_{i_{l+t}} y^*_{i_{l+t}} - \bar{x}_{i_{l+t}} \bar{y}_{i_{l+t}}, \Delta + y^*_{i_{l+t}} - \bar{y}_{i_{l+t}}) \\ & - \alpha_{i_{l+t}} (x^*_{i_{l+t}} - \bar{x}_{i_{l+t}}) - \beta_{i_{l+t}} (y^*_{i_{l+t}} - \bar{y}_{i_{l+t}}).\end{aligned}$$

Since we choose the coefficients $\alpha_{i_{l+t}}$ and $\beta_{i_{l+t}}$ such that

$$\alpha_{i_{l+t}} \phi_1 + \beta_{i_{l+t}} \phi_2 \leq \varphi_l(\phi_1 \phi_2 - \bar{x}_{i_{l+t}} \bar{y}_{i_{l+t}}, \phi_2 - \bar{y}_{i_{l+t}}),$$

and because $\varphi_l(\delta, \Delta) \leq \Upsilon_{l+t-1}(\delta, \Delta)$ by induction hypothesis, it follows that

$$\begin{aligned}\Upsilon_{l+t}(\delta, \Delta) &\geq \varphi_l(\delta + x^*_{i_{l+t}} y^*_{i_{l+t}} - \bar{x}_{i_{l+t}} \bar{y}_{i_{l+t}}, \Delta + y^*_{i_{l+t}} - \bar{y}_{i_{l+t}}) \\ &\quad - \varphi_l(x^*_{i_{l+t}} y^*_{i_{l+t}} - \bar{x}_{i_{l+t}} \bar{y}_{i_{l+t}}, y^*_{i_{l+t}} - \bar{y}_{i_{l+t}}) \\ &\geq \varphi_l(\delta, \Delta),\end{aligned}$$

where the last inequality is due to superadditivity of $\varphi_l(\cdot, \cdot)$. □

Thus we have shown that $\Upsilon_l(\cdot, \cdot)$ may be replaced by a weaker superadditive function $\varphi_l(\cdot, \cdot)$ to obtain valid lifting coefficients for the remaining variables. An obvious, almost trivial, choice for underestimator is perhaps the triangulation of the perturbation function over its domain. Note that superadditivity is required only over the set $\mathcal{R}_l$, which is a simplex formed by the three extreme points

1. $(\delta^1, \Delta^1) := (-\bar{q}_l, -\bar{p}_l)$,
2. $(\delta^2, \Delta^2) := (-\bar{q}_l, U_l - \bar{p}_l)$, and

3. $(\delta^3, \Delta^3) := (U_l - \bar{q}_l, U_l - \bar{p}_l)$,

where $\bar{p}_l = \sum_{i \in N \setminus L} \bar{y}_i$ and $\bar{q}_l = \sum_{i \in I \setminus L} \bar{x}_i \bar{y}_i$. Hence every point $(\delta, \Delta) \in \mathcal{R}_l$ can be written as

$$\begin{pmatrix} \delta \\ \Delta \end{pmatrix} = \left(1 - \frac{\Delta + \bar{p}_l}{U_l}\right) \begin{pmatrix} \delta^1 \\ \Delta^1 \end{pmatrix} + \frac{\Delta + \bar{p}_l - \delta - \bar{q}_l}{U_l} \begin{pmatrix} \delta^2 \\ \Delta^2 \end{pmatrix} + \frac{\delta + \bar{q}_l}{U_l} \begin{pmatrix} \delta^3 \\ \Delta^3 \end{pmatrix}.$$

Triangulating $\Upsilon_l(\cdot, \cdot)$ over $\mathcal{R}_l$ gives a affine function $\tilde{\varphi}_l(\delta, \Delta) := r_1 \delta + r_2 \Delta + r_0$, where

$$\begin{aligned} r_1 &= \frac{\Upsilon_l(\delta^3, \Delta^3) - \Upsilon_l(\delta^2, \Delta^2)}{U_l} \\ r_2 &= \frac{\Upsilon_l(\delta^2, \Delta^2) - \Upsilon_l(\delta^1, \Delta^1)}{U_l} \\ r_0 &= \frac{(U_l - \bar{p}_l)\, \Upsilon_l(\delta^1, \Delta^1) \;+\; (\bar{p}_l - \bar{q}_l)\, \Upsilon_l(\delta^2, \Delta^2) \;+\; \bar{q}_l\, \Upsilon_l(\delta^3, \Delta^3)}{U_l}. \end{aligned} \tag{48}$$

$\tilde{\varphi}_l(\cdot, \cdot)$ may not necessarily underestimate $\Upsilon_l(\cdot, \cdot)$ over $\mathcal{R}_l$. The actual conditions under which $\tilde{\varphi}_l(\cdot, \cdot) \leq \Upsilon_l(\cdot, \cdot)$ holds true depend on the seed inequality used for lifting and its associated perturbation function. If these conditions hold true, then by Jensen's inequality, $\tilde{\varphi}_l(\cdot, \cdot)$ must be the convex envelope of $\Upsilon_l(\cdot, \cdot)$ over $\mathcal{R}_l$. Suppose that these conditions hold true. Observe that $\Upsilon_l(0, 0) = 0$ since there exists at least one point on the seed inequality (either secant or gradient) that also belongs to the restriction $\mathcal{F}(N_1, N_2)$. Then, $\tilde{\varphi}_l(0, 0) = r_0 \leq \Upsilon_l(0, 0) = 0$. Since $\tilde{\varphi}_l(\delta, \Delta)$ is a affine function, superadditivity of $\tilde{\varphi}_l(\delta, \Delta)$ is equivalent to $r_0 \leq 0$. Now consider lifting a pair of variables $(x_{i_{l+1}}, y_{i_{l+t}})$ that have been fixed to $(\bar{x}_{i_{l+t}}, 0)$. We must find coefficients $\alpha_{i_{l+t}}$ and $\beta_{i_{l+t}}$ such that

$$\begin{aligned} & & \alpha_{i_{l+t}}(\phi_1 - \bar{x}_{i_{l+t}}) + \beta_{i_{l+t}} \phi_2 &\leq \tilde{\varphi}_l(\phi_1 \phi_2, \phi_2), \quad \forall \phi_1 \in \{0, 1\}, \phi_2 \in (0, u_{i_{l+t}}] \\ \phi_1 = \bar{x}_{i_{l+t}} \Rightarrow & & \beta_{i_{l+t}} \phi_2 &\leq r_2 \phi_2 + r_0, \qquad \forall \phi_2 \in (0, u_{i_{l+t}}] \\ \Rightarrow & & \beta_{i_{l+t}} &\leq r_2 + \frac{r_0}{\phi_2}, \qquad \forall \phi_2 \in (0, u_{i_{l+t}}], \end{aligned}$$

which implies a finite $\beta_{i_{l+t}}$ exists if and only if $r_0 \geq 0$. Similarly, we can show $r_0 \geq 0$ is required when $y_{i_{l+t}}$ has been fixed to $u_{i_{l+t}}$. Since we already argued that r_0 is always nonpositive, we must have $r_0 = 0$. If $r_0 < 0$, we translate up the perturbation function by $-r_0$. This translation of $\Upsilon_l(\cdot, \cdot)$ is equivalent to translating the seed inequality by $-r_0$. Hence if $r_0 < 0$, using $\tilde{\varphi}_l(\cdot, \cdot)$ as a lifting function requires lifting a weaker seed inequality

$\alpha_0 x_0 + \beta_j y_j \leq \gamma - r_0$. The translated lifting function then becomes a linear function $r_1\delta + r_2\Delta$. We next propose valid inequalities obtained using this linear underestimator.

Proposition 2.12. *Suppose that* (40) *is a valid inequality to* $\mathcal{P}(N \setminus L)$ *and let* $\tilde{\varphi}_l(\cdot,\cdot)$ *be a linear function such that* $\tilde{\varphi}_l(\delta,\Delta) = r_1\delta + r_2\Delta \leq \Upsilon_l(\delta,\Delta) - r_0$, *for all* $(\delta,\Delta) \in \mathcal{R}_l$, *where* r_1, r_2, r_0 *are given by* (48). *Define* $\tau \in \Re^{|N|}$ *as*

$$\tau_i := \begin{cases} r_1 + r_2 & i \in N_1 \setminus L \\ r_2 & i \in N_1^- \setminus L \\ r_1 a_{i-n} + r_2 & i \in N \setminus (I \cup L). \end{cases}$$

Then the following inequality is valid to the convex hull of $\mathcal{P}$.

$$\begin{aligned} \alpha_0 x_0 \;&+ \sum_{i \in L \cap I} \alpha_i (x_i - \bar{x}_i) \;+\; \beta_j y_j \;+ \sum_{i \in L \setminus j} \beta_i (y_i - \bar{y}_i) \\ &+ \sum_{i \in N_1 \setminus L} [u_i r_1]^+ (x_i - 1) \;+ \sum_{i \in N_1^- \setminus L} [u_i r_1]^- x_i \\ &+ \sum_{i \in N_2 \setminus L} \tau_i (y_i - u_i) \;+ \sum_{i \in N_2^- \setminus L} \tau_i y_i \;\leq\; \gamma - r_0. \end{aligned} \tag{49}$$

Proof. We wish to lift the weaker seed inequality $\alpha_0 x_0 + \beta_j y_j \leq \gamma - r_0$ by using the function $r_1\delta + r_2\Delta$ to find coefficients $\alpha_{i_{l+t}}$ and $\beta_{i_{l+t}}$, for all $i_{l+t} \notin L$, such that for all $\phi_1 \in \{0,1\}, \phi_2 \in [0, u_{i_{l+t}}]$ we have

$$\begin{aligned} \alpha_{i_{l+t}}(\phi_1 - \bar{x}_{i_{l+t}}) + \beta_{i_{l+t}}(\phi_2 - \bar{y}_{i_{l+t}}) &\leq r_1(\phi_1\phi_2 - \bar{x}_{i_{l+t}}\bar{y}_{i_{l+t}}) + r_2(\phi_2 - \bar{y}_{i_{l+t}}), \quad i_{l+t} \in I \setminus L \\ \beta_{i_{l+t}}(\phi_2 - \bar{y}_{i_{l+t}}) &\leq r_1(a_{i_{l+t}-n}(\phi_2 - \bar{y}_{i_{l+t}})) + r_2(\phi_2 - \bar{y}_{i_{l+t}}), \quad i_{l+t} \in N \setminus (I \cup L). \end{aligned}$$

Let $(\bar{x}_{i_{l+t}}, \bar{y}_{i_{l+t}})$ be the fixed extremal values of a variable pair $(x_{i_{l+1}}, y_{i_{l+t}})$. By Theorem 2.1, $\bar{x}_{i_{l+t}} \in \{0,1\}$ and $\bar{y}_{i_{l+t}} \in \{0, u_{i_{l+t}}\}$. We consider two cases.

Case 1 : $\bar{y}_{i_{l+t}} = 0$.

Let $i_{l+t} \in I \setminus L$ and set $\phi_1 = \bar{x}_{i_{l+t}}$. This implies $\beta_{i_{l+t}}\phi_2 \leq r_1\bar{x}_{i_{l+t}}\phi_2 + r_2\phi_2$ for all $\phi_2 \in (0, u_{i_{l+t}}]$. Hence $\beta_{i_{l+t}} = r_1\bar{x}_{i_{l+t}} + r_2$. For $i_{l+t} \in N \setminus (I \cup L)$, a similar argument implies $\beta_{i_{l+t}} = r_1 a_{i_{l+t}-n} + r_2$. Now set $\phi_1 = 1 - \bar{x}_{i_{l+t}}$. Consequently, we require

$$\alpha_{i_{l+t}}(1 - 2\bar{x}_{i_{l+t}}) + (r_1\bar{x}_{i_{l+t}} + r_2)\phi_2 \leq r_1(1 - \bar{x}_{i_{l+t}})\phi_2 + r_2\phi_2, \quad \forall \phi_2 \in [0, u_{i_{l+t}}],$$

which implies that $\alpha_{i_{l+t}} = [u_{i_{l+t}} r_1]^+$, if $i_{l+t} \in N_1$, and $\alpha_{i_{l+t}} = [u_{i_{l+t}} r_1]^-$, otherwise.

Case 2 : $\bar{y}_{i_{l+t}} = u_{i_{l+t}}$.

Let $i_{l+t} \in I \setminus L$. Setting $\phi_1 = \bar{x}_{i_{l+t}}$ implies $\beta_{i_{l+t}}(\phi_2 - u_{i_{l+t}}) \leq r_1\bar{x}_{i_{l+t}}(\phi_2 - u_{i_{l+t}}) + r_2(\phi_2 - u_{i_{l+t}})$, for all $\phi_2 \in [0, u_{i_{l+t}})$ and hence $\beta_{i_{l+t}} = r_1\bar{x}_{i_{l+t}} + r_2$. For $\phi_1 = 1 - \bar{x}_{i_{l+t}}$, we get that for all $\phi_2 \in [0, u_{i_{l+t}}]$,

$$\alpha_{i_{l+t}}(1-2\bar{x}_{i_{l+t}}) + (r_1\bar{x}_{i_{l+t}} + r_2)(\phi_2 - u_{i_{l+t}}) \leq r_1(1-\bar{x}_{i_{l+t}})\phi_2 + r_2\phi_2 - (r_1\bar{x}_{i_{l+t}} + r_2)u_{i_{l+t}},$$

which simplifies to $\alpha_{i_{l+t}} = [u_{i_{l+t}}r_1]^+$, if $i_{l+t} \in N_1$, and $\alpha_{i_{l+t}} = [u_{i_{l+t}}r_1]^-$, otherwise, as in the previous case.

□

Although the above proposition provides explicit valid inequalities, these may be weak. This is apparent since the derivation of these inequalities involved translating the seed inequality and then using a underestimator to enable sequence independent lifting; both these steps contribute to the weakening of the original tight seed inequality. In the next section, we address cases that yield (almost) superadditive functions and hence potentially produce strong valid inequalities.

2.5.2 Valid inequalities from secants

We now use the preceding lifting theory to obtain valid inequalities to the convex hull of $\mathcal{P}$ by choosing $L = \{j\}$ and $l = 1$. Proposition 2.8 provides inequalities in the (x_0, y_j, x_j)-space obtained by lifting x_j in the seed inequality. (When $j \in N \setminus I$, there is no lifting step for x_j and we use the original seed inequality.) Thus, we lift variable pairs (x_i, y_i) that were fixed at $(\bar{x}_i, \bar{y}_i)$, for all $i \in N \setminus j$, into a inequality of the form $\alpha_0 x_0 + \alpha_j(x_j - \bar{x}_j) + \beta_j y_j \leq \gamma$. The associated perturbation function is

$$\begin{aligned} \Upsilon(\delta, \Delta) := \quad \gamma \; - \max \quad & \alpha_0 x_0 + \alpha_j(x_j - \bar{x}_j) + \beta_j y_j \\ \text{s.t.} \quad & x_j y_j + \bar{q} + \delta = x_0\,(y_j + \bar{p} + \Delta) \\ & y_j + \bar{p} + \Delta \;\leq\; u_0 \\ & x_0 \in [0,1],\; x_j \in \{0,1\}, y_j \in [0, u_j]. \end{aligned} \tag{50}$$

We only consider facets of trivial restrictions of $\mathcal{P}$ and compute their associated perturbation functions in closed form using Lemma 2.4. From Corollary 2.4, a restriction $\mathcal{F}(N_1, N_2)$

is trivial and has secant inequalities if and only if $\bar{p} = \bar{q} = 0$. Since $\bar{p} = \sum_{i \in N \setminus j} \bar{y}_i$ and $\bar{q} = \sum_{i \in I \setminus j} \bar{x}_i \bar{y}_i$ (cf. (29)), this trivial restriction corresponds to fixing $y_i = \bar{y}_i = 0$, for all $i \in N \setminus j$, denoted as $\mathcal{F}(N_1^-, N_1, N \setminus j, \emptyset)$. Thus we are interested in lifting pairs of variables (x_i, y_i) that have been fixed to either $(1, 0)$ or $(0, 0)$. Note that for this fixing, the domain of interest for checking superadditivity of $\Upsilon(\cdot, \cdot)$ is the set

$$\mathcal{R} = \{(\delta, \Delta) \colon 0 \leq \delta \leq \Delta \leq U_j\},$$

where $U_j = \min\{u_0, \sum_{k \in N \setminus j} u_k\}$.

For $j \in I$, the seed inequality is a secant (cf. Proposition 2.8).

$$\begin{aligned} -u_j x_0 + y_j + u_j x_j &\leq u_j \qquad j \in N_1 \\ u_j x_0 + y_j - u_j x_j &\leq u_j \qquad j \in N_1^-. \end{aligned}$$

Proposition 2.13. *Let $j \in N_1$ and consider the inequality $-u_j x_0 + y_j + u_j x_j \leq u_j$ valid to $\mathcal{F}(N_1^-, N_1, N \setminus j, \emptyset)$. Its perturbation function is*

$$\Upsilon(\delta, \Delta) = \begin{cases} \dfrac{u_j(\delta - \Delta)}{\Delta + u_j}, & 0 \leq \Delta \leq u_0 - u_j \\ u_j - u_0 + \dfrac{u_j}{u_0}\delta + \left(1 - \dfrac{u_j}{u_0}\right)\Delta, & u_0 - u_j < \Delta \leq U_j. \end{cases}$$

Moreover, $\Upsilon(\cdot, \cdot)$ is superadditive over $\mathcal{R}$.

For $j \in N_1^-$ and $u_j x_0 + y_j - u_j x_j \leq u_j$, the function is

$$\Upsilon(\delta, \Delta) = \begin{cases} \dfrac{-u_j \delta}{\Delta + u_j}, & 0 \leq \Delta \leq u_0 - u_j \\ u_j - u_0 - \dfrac{u_j}{u_0}\delta + \Delta, & u_0 - u_j < \Delta \leq U_j \end{cases}$$

which is also superadditive over $\mathcal{R}$.

Proof. Consider the case $j \in N_1$ and the lifting problem (50). Note that $x_j \in \{0, 1\}$. We use Φ_0 and Φ_1 to denote the optimal values for the maximization problem in (50) corresponding to $x_j = 0$ and $x_j = 1$, respectively. Thus, $\Upsilon(\delta, \Delta) = \gamma - \max\{\Phi_0, \Phi_1\}$. Also denote $\mu_j' = \min\{u_j, u_0 - \Delta\}$. When $x_j = 0$, since $u_j > 0$, Lemma 2.4 implies that the optimum is attained at $(\delta/(\Delta + \mu_j'), \mu_j')$ and the value is

$$\Phi_0 = \frac{-u_j \delta}{\Delta + \min\{u_j, u_0 - \Delta\}} + \min\{u_j, u_0 - \Delta\}.$$

For x_j fixed to 1, Lemma 2.4 tells us that the optimum can be attained at one of the two extreme points : $(\delta/\Delta, 0)$ or $((\delta+\mu_j^{'})/(\Delta+\mu_j^{'}), \mu_j^{'})$. Now the seed inequality is a secant joining $(0,0)$ and $(1,u_j)$ and with a slope of u_j. The secant drawn between $(\delta/\Delta, 0)$ and $((\delta+\mu_j^{'})/(\Delta+\mu_j^{'}), \mu_j^{'})$ has a slope equal to $\Delta(\Delta+\mu_j^{'})/(\Delta-\delta)$, if $\delta < \Delta$, and $+\infty$ otherwise. Now,

$$\frac{\Delta(\Delta+\mu_j^{'})}{\Delta-\delta} \;\geq\; u_j \quad \iff \quad \Delta(\Delta+\mu_j^{'}-u_j)+\delta u_j \;\geq\; 0,$$

which is true because $\mu_j^{'} = \min\{u_j, u_0-\Delta\}$, $u_j \leq u_0$, and $\delta, \Delta \geq 0$. Hence the secant drawn between $(\delta/\Delta, 0)$ and $((\delta+\mu_j^{'})/(\Delta+\mu_j^{'}), \mu_j^{'})$ has a greater slope, implying that the optimum must be attained at $((\delta+\mu_j^{'})/(\Delta+\mu_j^{'}), \mu_j^{'})$. This gives us

$$\Phi_1 = \frac{-u_j(\delta+\min\{u_j, u_0-\Delta\})}{\Delta+\min\{u_j, u_0-\Delta\}} + \min\{u_j, u_0-\Delta\} + u_j.$$

Then, $\Phi_1 \geq \Phi_0$ because

$$\Phi_1 - \Phi_0 = u_j\left[\frac{\Delta}{\Delta+\min\{u_j, u_0-\Delta\}}\right] \;\geq\; 0.$$

Hence, $\Upsilon(\delta,\Delta) = u_j - \Phi_1$ and the proposed closed form expression follows subsequently.

We make the following observation about $\Upsilon(\cdot,\cdot)$.

Observation 2.6. *For any Δ such that $u_0 - u_j < \Delta \leq U_j$,*

$$\frac{u_j(\delta-\Delta)}{\Delta+u_j} \;\leq\; u_j - u_0 + \frac{u_j}{u_0}\delta + \left(1-\frac{u_j}{u_0}\right)\Delta.$$

Proof. The statement is equivalent to showing that

$$\begin{aligned} \Delta + u_j - u_0 + u_j(\delta-\Delta)\left[\frac{1}{u_0} - \frac{1}{\Delta+u_j}\right] &\geq 0 \\ \iff \quad (\Delta+u_j-u_0)\left[1-\frac{u_j(\Delta-\delta)}{u_0(\Delta+u_j)}\right] &\geq 0 \end{aligned}$$

which is true since $\Delta+u_j > u_0$ by assumption and $0 \leq u_j/u_0 \leq 1$ and $0 \leq (\Delta-\delta)/(\Delta+u_j) \leq 1$. □

For checking superadditivity, first observe that if $\Delta > u_0 - u_j$, then $\Upsilon(\cdot,\cdot)$ is a affine function with a nonpositive constant term, and hence superadditive. Now take two points $(\delta_1,\Delta_1), (\delta_2,\Delta_2) \in \mathcal{R}$ such that $(\delta_1+\delta_2, \Delta_1+\Delta_2) \in \mathcal{R}$. We consider three cases.

$\Delta_1, \Delta_2, \Delta_1 + \Delta_2 \leq u_0 - u_j$ **:** The difference $\Upsilon(\delta_1 + \delta_2, \Delta_1 + \Delta_2) - \Upsilon(\delta_1, \Delta_1) - \Upsilon(\delta_2, \Delta_2)$ is given by

$$\begin{aligned} & u_j \left[\frac{\delta_1 + \delta_2 - \Delta_1 - \Delta_2}{\Delta_1 + \Delta_2 + u_j} - \frac{\delta_1 - \Delta_1}{\Delta_1 + u_j} - \frac{\delta_2 - \Delta_2}{\Delta_2 + u_j} \right] \\ = \; & u_j \left[(\delta_1 - \Delta_1) \left(\frac{1}{\Delta_1 + \Delta_2 + u_j} - \frac{1}{\Delta_1 + u_j} \right) \right. \\ & \qquad \left. + \; (\delta_2 - \Delta_2) \left(\frac{1}{\Delta_1 + \Delta_2 + u_j} - \frac{1}{\Delta_2 + u_j} \right) \right] \\ = \; & \frac{u_j}{\Delta_1 + \Delta_2 + u_j} \left[\frac{\Delta_2(\Delta_1 - \delta_1)}{\Delta_1 + u_j} + \frac{\Delta_1(\Delta_2 - \delta_2)}{\Delta_2 + u_j} \right], \end{aligned}$$

which is nonnegative since $(\delta_1, \Delta_1), (\delta_2, \Delta_2) \in \mathcal{R}$ implies $\Delta_1, \Delta_2, \Delta_1 - \delta_1, \Delta_2 - \delta_2 \geq 0$. In fact, it follows that $u_j(\delta - \Delta)/(\Delta + u_j)$ is superadditive over $\mathcal{R}$.

$\Delta_1, \Delta_2 \leq u_0 - u_j, \Delta_1 + \Delta_2 > u_0 - u_j$ **:** This case holds true because of Observation 2.6 and the previous case which proved the superadditivity of $u_j(\delta - \Delta)/(\Delta + u_j)$ over $\mathcal{R}$.

$\Delta_1 \leq u_0 - u_j, \Delta_2 > u_0 - u_j \;\Rightarrow\; \Delta_1 + \Delta_2 > u_0 - u_j$ **:**

$$\Upsilon(\delta_1 + \delta_2, \Delta_1 + \Delta_2) - \Upsilon(\delta_1, \Delta_1) - \Upsilon(\delta_2, \Delta_2) \;=\; \frac{u_j \delta_1}{u_0} + \left(1 - \frac{u_j}{u_0}\right) \Delta_1 + \frac{u_j(\Delta_1 - \delta_1)}{\Delta_1 + u_j}$$

which is again nonnegative due to $u_j \leq u_0$ and $\Delta_1 \geq \delta_1$.

For $j \in N_1^-$, the arguments are similar and we only provide a sketch of the proof. We first derive that

$$\begin{aligned} \Phi_0 &= \frac{u_j \delta}{\Delta + \min\{u_j, u_0 - \Delta\}} + \min\{u_j, u_0 - \Delta\} \\ \Phi_1 &= \frac{u_j(\delta + \min\{u_j, u_0 - \Delta\})}{\Delta + \min\{u_j, u_0 - \Delta\}} + \min\{u_j, u_0 - \Delta\} - u_j. \end{aligned}$$

Observe that in this case $\Phi_0 \geq \Phi_1$ and hence $\Upsilon(\delta, \Delta) = u_j - \Phi_0$. For checking superadditivity, in the first case, the difference is $(u_j/(\Delta_1 + \Delta_2 + u_j))(\delta_1 \Delta_2/(\Delta_1 + u_j) + \delta_2 \Delta_1/(\Delta_2 + u_j))$, which is nonnegative, and in the third case it is $\Delta_1 + u_j \delta_1 (1/(\Delta_1 + u_j) - 1/u_0) \geq 0$, since $\Delta_1 \leq u_0 - u_j$. The second case holds due to an analogue of Observation 2.6. □

We next illustrate these perturbation functions in Figure 7 for the following example.

Example 2.2. Let $n = 3, m = 2$ and $\mathcal{P}$ be the set

$$\mathcal{P} = \Big\{(x,y) \in \Re^4_+ \times \Re^5_+ \colon x_1y_1 + x_2y_2 + x_3y_3 + 1/2y_4 + 1/3y_5 = x_0(y_1 + y_2 + y_3 + y_4 + y_5)$$
$$y_1 + y_2 + y_3 + y_4 + y_5 \leq 60$$
$$x \in [0,1]^4, y_1 \leq 10, y_2 \leq 14, y_3 \leq 17, y_4 \leq 22, y_5 \leq 30\Big\}.$$

Choose $j = 3$ and consider the secant for $j \in N_1$

$$-17x_0 + y_3 + 17x_3 \quad \leq \quad 17 \tag{51}$$

Here $\mathcal{R} = \{(\delta, \Delta) \colon 0 \leq \delta \leq \Delta \leq 76\}$. The function is nonlinear for $\Delta \in [0, 43]$ and affine otherwise.

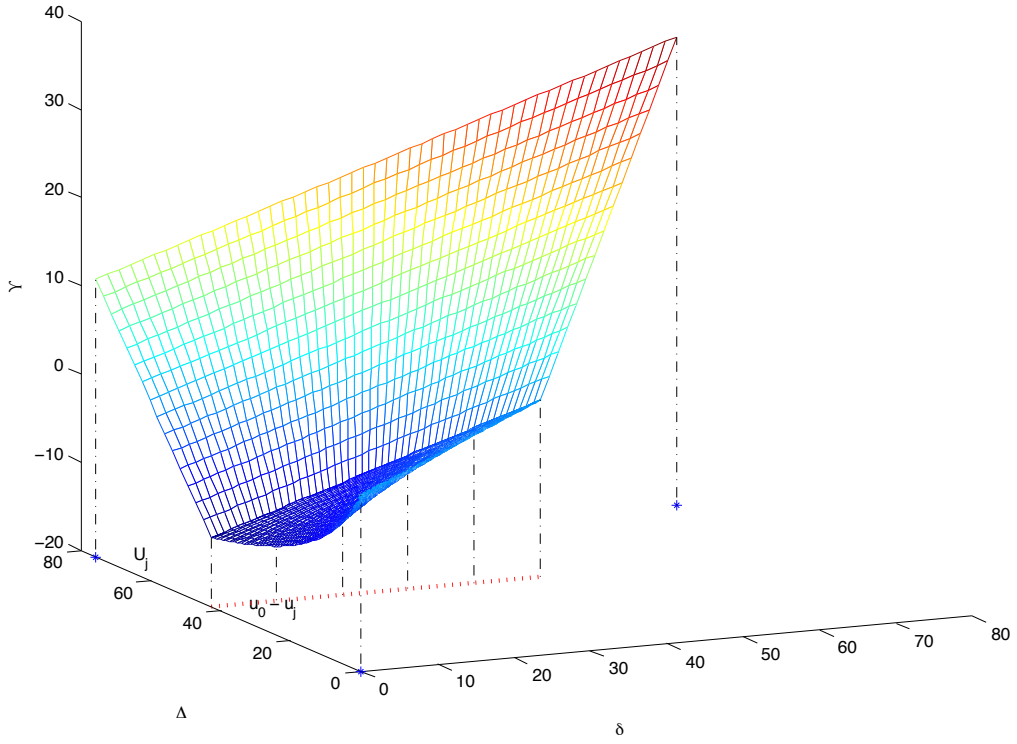


Figure 7: Perturbation function from Proposition 2.13 applied to Example 2.2. Seed inequality is (51).

Because $\Upsilon(\cdot,\cdot)$ is superadditive over $\mathcal{R}$, we can directly apply the result of Proposition 2.10 to obtain valid inequalities for $\mathcal{P}$.

Theorem 2.2. *For every $j \in I$ and $N_1 \subseteq I \setminus j$ with $N_1^- = I \setminus (N_1 \cup j)$, the following two inequalities are valid to* $\mathrm{conv}(\mathcal{P})$.

$$-u_j x_0 + y_j + u_j x_j \; + \sum_{i \in N_1} \frac{u_j \sigma_i}{u_j + \sigma_i} x_i \; - \sum_{i \in N_1^-} y_i \; - \sum_{i \in N \setminus I} (1 - a_{i-n}) y_i \;\; \leq \;\; u_j + \sum_{i \in N_1} \frac{u_j \sigma_i}{u_j + \sigma_i}$$

$$u_j x_0 + y_j - u_j x_j \; - \sum_{i \in N_1^-} \frac{u_j \sigma_i}{u_j + \sigma_i} x_i \; - \sum_{i \in N_1} y_i \; - \sum_{i \in N \setminus I} a_{i-n} y_i \;\; \leq \;\; u_j,$$

where $\sigma_i = \min\{u_i, u_0 - u_j\}$ for all $i \in I \setminus j$.

Proof. Consider the first seed inequality $-u_j x_0 + y_j + u_j x_j \leq u_j$. Let $i \in N \setminus j$. We are required to find coefficients α_i, β_i such that they satisfy (44) and (45) with respect to $\Upsilon(\cdot, \cdot)$. For $i \in I$, we want α_i and β_i such that $\alpha_i(\phi_1 - \bar{x}_i) + \beta_i \phi_2 \leq \Upsilon(\phi_1 \phi_2, \phi_2)$, for all $\phi_1 \in \{0, 1\}, \phi_2 \in [0, u_i]$. Similarly for $i \notin I$. Since the definition of $\Upsilon(\cdot, \cdot)$ varies over two partitions of $\mathcal{R}$, namely $\{\phi_2 \leq u_0 - u_j\} \vee \{\phi_2 > u_0 - u_j\}$, and ϕ_2 takes all positive values upto u_i, we must consider two separate cases depending on the upper bound u_i.

$u_i + u_j \leq u_0$ **:** Here $\Upsilon(\phi_1 \phi_2, \phi_2) = u_j(\phi_1 - 1)\phi_2/(\phi_2 + u_j)$. First let $i \in N_1$. Fix $\phi_1 = 1$. Then $\beta_i \leq 0$; choose $\beta_i = 0$. Now fixing $\phi_1 = 0$ implies $\alpha_i \geq \sup_{[0, u_i]} u_j \phi_2/(u_j + \phi_2)$. This function is nondecreasing and hence the supremum is achieved at $\phi_2 = u_i$, which gives us $\alpha_i \geq u_j u_i/(u_j + u_i)$. Now let $i \in N_1^-$. Fixing $\phi_1 = 0$ implies $\beta_i \leq \inf_{(0, u_i]} -u_j/(u_j + \phi_2)$. This infimum is attained at 0 and hence $\beta_i \leq -1$. Then $\phi_1 = 1$ implies $\alpha_i - \phi_2 \leq 0$ and hence $\alpha_i \leq 0$. Finally let $i \in N \setminus I$. Here we only have to lift y_i. We need $\beta_i \phi_2 \leq u_j(a_{i-n} - 1)\phi_2/(u_j + \phi_2)$ which simplifies to $\beta_i \leq \inf_{(0, u_i]} u_j(a_{i-n} - 1)/(u_j + \phi_2) = a_{i-n} - 1$.

$u_i + u_j > u_0$ **:** The perturbation function is

$$\Upsilon(\phi_1 \phi_2, \phi_2) = \begin{cases} \dfrac{u_j \phi_2(\phi_1 - 1)}{\phi_2 + u_j}, & 0 \leq \phi_2 \leq u_0 - u_j \\ \\ u_j - u_0 + \phi_2 + \dfrac{u_j}{u_0} \phi_2(\phi_1 - 1), & u_0 - u_j < \phi_2 \leq u_i. \end{cases}$$

Here we must optimize separately over the two partitions of $\mathcal{R}$. Let $j \in N_1$ and fix $\phi_1 = 1$. Then $\beta_i \leq \min\{0,\; 1 + \inf_{(u_0 - u_j, u_i]}(u_j - u_0)/\phi_2\} = 0$. Now fixing $\phi_1 = 0$ gives

$$-\alpha_i \leq \min\left\{ \inf_{(0, u_0 - u_j]} -u_j \phi_2/(u_j + \phi_2),\; \inf_{(u_0 - u_j, u_i]} u_j - u_0 + (1 - u_j/u_0)\phi_2 \right\}.$$

It is readily checked that the two infimums are attained at $u_0 - u_j$, giving $\alpha_i \geq u_j(u_0 - u_j)/u_0$. Now let $i \in N_1^-$. For $\phi_1 = 0$,

$$\beta_i \leq \min\left\{\inf_{(0,u_0-u_j]} -u_j/(u_j+\phi_2),\ (1-u_j/u_0) + \inf_{(u_0-u_j,u_i]}(u_j-u_0)/\phi_2\right\}.$$

The two infimums are at 0 and $u_0 - u_j$, respectively. Hence, $\beta_i \leq \min\{-1, -u_j/u_0\} = -1$ since $u_j \leq u_0$. This implies $\alpha_i \leq \min\{\inf_{(0,u_0-u_j]} \phi_2,\ \inf_{(u_0-u_j,u_i]} u_j - u_0 + 2\phi_2\} = \min\{0, u_0 - u_j\} = 0$. Finally $i \in N \setminus I$. Here

$$\begin{aligned}\beta_i &\leq \min\left\{\inf_{(0,u_0-u_j]} u_j(a_{i-n}-1)/(u_j+\phi_2),\ u_j(a_{i-n}-1)/u_0 + 1 + \inf_{(u_0-u_j,u_i]}(u_j-u_0)/\phi_2\right\}\\ &= \min\{a_{i-n}-1, u_j(a_{i-n}-1)/u_0\}\\ &= a_{i-n}-1\end{aligned}$$

since $u_j \leq u_0$ and $a_{i-n} \leq 1$.

The proof for the second valid inequality lifted from $u_j x_0 + y_j - u_j x_j \leq u_j$ is identical by symmetry. □

2.6 Conclusion

In this chapter, we have studied a continuous bilinear set that commonly arises at a single node in a network flow problem. We characterized the extreme points of this set and provided sufficient conditions under which its convex hull is polyhedral. We analyzed and compared the strengths of standard polyhedral relaxations that arise from envelopes of a bilinear term. Under certain assumptions, it was shown that the convex hull of a variant of our set is given by its envelope-based relaxation. Our main contribution was in deriving linear inequalities in the original space valid for our set. Towards this end, we first studied restrictions by fixing variables at their extreme values. These restrictions yielded a conic quadratic representation for our set and a disjunctive polyhedral relaxation. The classical lifting theory was extended to enable lifting these restrictions. An exponential class of valid inequalities was produced. To the best of our knowledge, this is the first study that contributes a polyhedral relaxation in original space for this set. Since our inequalities were shown to satisfy strong lifting properties, we also expect this relaxation to be stronger than conventional higher-dimensional relaxations.

2.7 Notes

In this section, we present a sufficient condition for polyhedrality of conv($\mathcal{Q}$). Due to Lemma 2.1, the cases remaining to be addressed are those where

1. $x_j \in (\tilde{l}_j, \tilde{u}_j)$ and $y_j \in (l_j, u_j)$, or
2. $x_j \in (\tilde{l}_j, \tilde{u}_j)$, or
3. $y_j \in (l_j, u_j)$,

for some j and all other variables are at one of their bounds. The answer to whether points belonging to one of these two cases are extreme points of conv($\mathcal{Q}$) depends on the values of the coefficients and the bounds on the variables.

Example 2.3. $\mathcal{Q} = \{(x, y) \in \Re^2 \times \Re^2 \colon x_1 y_1 - x_2 y_2 = 2, -1 \leq x_1, y_1 \leq 5, -1 \leq x_2, y_2 \leq 2\}$. Fix $x_2 = y_2 = 2$, so that $x_1 y_1 = 6$ for (x, y) to be feasible. Then every point with $y_1 = 6/x_1$ for some $x_1 \in [\frac{6}{5}, 5]$ is an extreme point of the convex hull of $\mathcal{Q}$.

When the variables are symmetric about the origin, the situation from the above example does not occur.

Proposition 2.14. *If* $\tilde{l}_i = -\tilde{u}_i, l_i = -u_i$, *for all* $i = 1, \ldots, n_1 + n_2 + m_1 + m_2$, *then* conv($\mathcal{Q}$) *is a polyhedral set.*

Proof. Consider a point $(x, y) \in \mathcal{Q}$. If two variables x_j and x_k take non-extreme values, then we have shown in Lemma 2.1 that (x, y) cannot be an extreme point.

Now suppose that $x_1 \in (\tilde{l}_1, \tilde{u}_1)$ and $y_1 \in (l_1, u_1)$ and all other variables are at one of their bounds. Let $c_1 x_1 y_1 = b - \sigma$. Since $c_1 > 0$, we can rewrite $x_1 y_1 = \frac{b-\sigma}{c_1}$. Suppose, for the sake of illustration that $b - \sigma > 0$. Consider Figure 8.

Let $\Delta_1 = \frac{b-\sigma}{c_1 u_1}$ and $\tilde{\Delta}_1 = \frac{b-\sigma}{c_1 \tilde{u}_1}$. The coordinates of A,B,and C are given by (Δ_1, u_1), $(\tilde{\Delta}_1, \tilde{u}_1)$, and (x_1, y_1), respectively. Point D is obtained by intersecting the line joining the origin to C with the segment AB. Points F,G,E,H are reflections of the points A,B,C,D about the origin. Thus, the points C and E lie in the interior of the segment DH. Since D and H lie on segments AB and FG, respectively, they belong to conv($\mathcal{Q}$). This implies

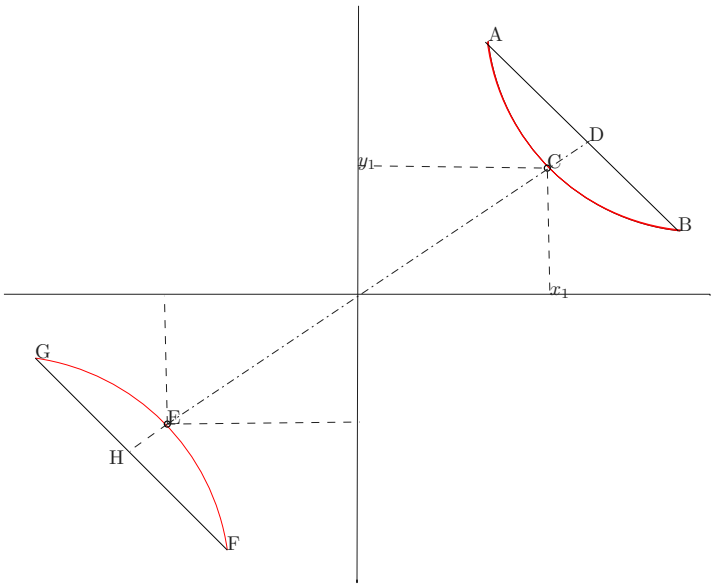


Figure 8: Symmetric bounds on variables in (21).

that our chosen point C cannot be an extreme point of conv($\mathcal{Q}$). If $b - \sigma < 0$, then the discussion is similar with the only difference from Figure 8 being that the points are in the second and fourth quadrant. Note that if $b - \sigma = 0$, then C and E both coincide with the origin.

There may exist an extreme point of conv($\mathcal{Q}$) such that only $x_j \in (\tilde{l}_j, \tilde{u}_j)$ for some index j and all other variables are at one of their bounds, for example points A,B,F,G in Figure 8. This point may belong to $\mathcal{Q}$ perhaps for any combination of remaining variables set to their upper and lower bounds, and in total, there are only finitely many extreme points of this nature. □

CHAPTER III

MILP APPROACHES TO MIXED INTEGER BILINEAR PROGRAMMING

The pooling problem was introduced as a continuous bilinear program (BLP) in Chapter 1. Restrictions of this problem provide an upper bound on the global optimal value. One way of obtaining restrictions of a BLP is to discretize one set of variables within their respective bounds, thus leading to a mixed integer bilinear program (MIBLP). In this MIBLP, each bilinear term is a product of a nonnegative integer variable and a nonnegative continuous variable. This chapter studies mixed integer linear programming (MILP)-based solution methodologies for solving a general MIBLP.

3.1 Introduction

Consider a mixed integer bilinear program given as

$$\begin{aligned} \min_{x,y} \quad & x^\top Q_0 y + f_0^\top x + g_0^\top y \\ \text{s.t.} \quad & Ax + Gy \leq h_0 \\ & x^\top Q_t y + f_t^\top x + g_t^\top y \leq h_t, \qquad t = 1, \ldots, p, \\ & \mathbf{0} \leq x \leq \tilde{u} \\ & \mathbf{0} \leq y \leq u,\ y \in \mathbb{Z}^n, \end{aligned} \tag{MIBLP1}$$

where $Q_t \in \Re^{m \times n}, f_t \in \Re^m, g_t \in \Re^n$, for $t = 0, \ldots, p$, and $A \in \Re^{q \times m}, G \in \Re^{q \times n}, h_0 \in \Re^q$. We assume that all variables have a lower bound of zero. Finite upper bounds on x and y are given by $\tilde{u} \in \Re^m_+$ and $u \in \Re^n_+$, respectively. In the formulation (MIBLP1), every bilinear term is a product of one continuous variable x_l and one integer variable y_j. Equality constraints, if present, are represented by two inequalities of $\leq$-type.

Continuous and mixed integer bilinear problems find many applications [29, 56, 64, 72, 74, 85, 90] and have been fairly well studied in literature. A common solution methodology is to construct polyhedral relaxations using envelopes of each bilinear term [5] within a

spatial branch-and-bound framework [43]. Tighter relaxations can be constructed using convex envelopes of the entire bilinear function [103, 105]. There also exist specialized branch-and-bound algorithms that contract the feasible region at each node of the search tree [117]. The reformulation linearization technique (RLT) of Sherali and Adams [97] has been applied to the continuous bilinear problem [98] and extended to the mixed $\{0,1\}$ problem with a bilinear objective function [2]. The branch-and-cut algorithm in [14] uses four classes of RLT inequalities to solve a pooling problem. Convex relaxations based on semidefinite programming have been studied [11]. Another type of relaxation is based on Lagrangian duals [3, 25, 42]. One may also obtain piecewise linear relaxations by dividing the intervals of one or both the variables in a bilinear term into sufficient number of pieces and constructing envelopes in each of these sub-intervals [57]. Branching strategies [23] and heuristics [35] have also been developed.

The main objective of this study is to seek MILP-based solution approaches using polyhedral study of single term bilinear sets. A MILP-based methodology can be particularly advantageous if, besides the nonconvexities of bilinear terms, the integrality constraints on variables are "hard" to satisfy. Since considerable progress has been made in algorithms and state-of-the-art solvers for MILP, these hard constraints can be better dealt with through a MILP solution procedure. The proposed approach differs from previous work in that our focus is on solving (MIBLP1) as a MILP whereas the existing methods are aimed at obtaining stronger relaxations, branching techniques, and heuristics within a spatial branch-and-bound framework for solving (MIBLP1). Hence, our first step is to use binary expansion of general integer variables to obtain an extended reformulation. Although the use of binary expansions is known to be inefficient for general MILPs [81], in our case, it gives us an exact MILP reformulation of (MIBLP1). On the contrary, the use of McCormick envelopes (cf. (13)) produces a relaxation of the mixed integer bilinear term. Binary expansions were proposed by [49, 56] for reformulating mixed integer bilinear sets. However, to the best of our knowledge there has been no study of the polyhedral structure of the sets arising due to such binary reformulations. Our contribution is to obtain complete descriptions of the convex hulls of these reformulated single term bilinear sets and use them

in a branch-and-cut algorithm for solving the reformulated MILP.

The rest of the chapter is organized as follows. §3.2 discusses MILP formulations for (MIBLP1) and studies their relative strengths. Of these two formulations, one is a relaxation obtained using McCormick envelopes of $w_{lj} = x_l y_j$ and the other is a reformulation due to binary representation of y_j. In §3.3, we study the single term mixed integer bilinear set. We derive all the facet-defining inequalities of the convex hull of this set. In §3.4, we present some computational results to demonstrate the effectiveness of our cuts. Next, §3.5 presents an exponential family of valid inequalities for the single term mixed integer bilinear set with a nontrivial upper bound on the bilinear term. Finally in §3.6, we consider reformulating each y_j using an arbitrary natural number and generalize our inequalities from the binary expansion approach.

We use the following notation in this chapter : $\text{conv}(\cdot)$ is the convex hull of a set and $\text{relax}(\cdot)$ is the continuous relaxation of a set obtained by dropping the integrality restrictions on its variables. $\text{Proj}_x(\cdot)$ is the projection of a set onto the x-space. For ease of notation, we sometimes represent a singleton $\{i\}$ simply as i. $\Re_+$ is the set of nonnegative reals, and $\mathbb{Z}_+$ and $\mathbb{Z}_{++}$ are the set of nonnegative and positive integers, respectively.

3.2 MILP formulations

Let us linearize the objective function and constraints in (MIBLP1) by introducing new variables $w_{lj} = x_l y_j$, for all $l \in \{1, ..., m\}$, $j \in \{1, ..., n\}$. This gives us the following reformulation (MIBLP) in an extended space.

$$
\begin{aligned}
\min \quad & \sum_{l=1}^{m} \sum_{j=1}^{n} Q_{0lj} w_{lj} + f_0^\top x + g_0^\top y \\
\text{s.t.} \quad & Ax + Gy \leq h_0 \\
& \sum_{l=1}^{m} \sum_{j=1}^{n} Q_{tlj} w_{lj} + f_t^\top x + g_t^\top y \leq h_t, \qquad t = 1, \ldots, p, \\
& w_{lj} = x_l y_j, \quad l = 1, \ldots, m,\ j = 1, \ldots, n \\
& \mathbf{0} \leq x \leq \tilde{u}, \\
& \mathbf{0} \leq y \leq u,\ y \in \mathbb{Z}^n.
\end{aligned}
\tag{MIBLP}
$$

Note that in the above reformulation we have reduced all the bilinearities to the constraints $w_{lj} = x_l y_j$, for all $l \in \{1, ..., m\}$, and $j \in \{1, ..., n\}$. In the absence of these bilinearities, the problem is a MILP. Hence, solving (MIBLP) using MILP techniques is possible only if we obtain MILP reformulations of the bilinear terms. To do this, we study reformulations of each bilinear term separately.

3.2.1 Reformulations of single term mixed integer bilinear set

In this subsection, we study a single bilinear term $w = xy$ abstracted from (MIBLP). For notational convenience, we drop the subscripts on the variables (x_l, y_j, w_{lj}) and denote them simply as (x, y, w).

For bounded continuous and general integer variables x and y, respectively, and a bilinear variable $w = xy$, consider the mixed integer bilinear set:

$$\mathcal{X} := \{(x, y, w) \in \Re_+ \times \mathbb{Z}_+ \times \Re \colon w = xy, x \leq a, y \leq b\}. \tag{52}$$

We assume that $b \geq 1$ is a positive integer and $a > 0$ is a positive real. This set incorporates the individual bounds on x and y, but does not include nontrivial bounds (if any) on w. A standard approach for linearizing such bilinear terms is to replace each term by its convex and concave envelopes, also called the McCormick envelopes (cf. (13)). Performing this operation on $\mathcal{X}$ gives us the following set

$$\mathcal{M} := \{(x, y, w) \in \Re \times \mathbb{Z}_+ \times \Re \colon w \geq 0, w \leq ay, w \leq bx, w \geq bx + ay - ab\}. \tag{53}$$

Another idea is to use a unary or binary expansion of the integer variable y. Let z be the new binary vector used in such an expansion. Using v_i to model the product xz_i for each i, we obtain the sets

$$\mathcal{U} := \Big\{(x, y, w, z, v) \in \Re \times \mathbb{Z}_+ \times \Re \times \{0,1\}^b \times \Re^b \colon y = \sum_{i=1}^{b} i z_i, \sum_{i=1}^{b} z_i \leq 1, w = \sum_{i=1}^{b} i v_i, \\ v_i \geq 0, v_i \leq a z_i, v_i \leq x, v_i \geq x + a z_i - a, \forall i \in \{1, \ldots, b\}\Big\}, \tag{54}$$

for unary expansion and

$$\mathcal{B} := \Big\{(x, y, w, z, v) \in \Re \times \mathbb{Z}_+ \times \Re \times \{0,1\}^k \times \Re^k \colon y = \sum_{i=1}^{k} 2^{i-1} z_i \ \leq b,\ w = \sum_{i=1}^{k} 2^{i-1} v_i,$$
$$v_i \geq 0, v_i \leq a z_i, v_i \leq x, v_i \geq x + a z_i - a, \forall i \in \{1, \ldots, k\}\Big\}, \tag{55}$$

for binary expansion, where $k = \lfloor \log_2 b \rfloor + 1$. The lower and upper bounds on x and y are implied in each of the above three formulations. Note that for $\mathcal{U}$ and $\mathcal{B}$, the linearization of $v_i = x z_i$ is exact because $z_i \in \{0, 1\}$, for all i. We first compare the strengths of these sets in the following result.

Proposition 3.1. *$\mathcal{X} = \text{Proj}_{x,y,w}(\mathcal{U}) = \text{Proj}_{x,y,w}(\mathcal{B})$ and $\mathcal{X} \subseteq \mathcal{M}$. The set $\mathcal{M} \setminus \mathcal{X}$ is nonempty if and only if $b \geq 2$.*

Proof. By construction, it follows that $\mathcal{X} \subseteq \mathcal{M}$, $\mathcal{X} \subseteq \text{Proj}_{x,y,w}(\mathcal{U})$, and $\mathcal{X} \subseteq \text{Proj}_{x,y,w}(\mathcal{B})$. We prove the reverse inclusion only for $\mathcal{U}$. The proof for $\mathcal{B}$ is similar. Consider any feasible point $(x, y, w, z, v) \in \mathcal{U}$. Since $y \in \mathbb{Z}_+$, there are two cases -

1. $y = 0$. Then $z_i = 0$, for all $i \in \{1, \ldots, b\}$, which implies that $v_i = 0$, for all $i \in \{1, \ldots, b\}$. Therefore $w = yx$.

2. $y > 0$. Then $z_y = 1$ and $z_i = 0, i \in \{1, ..., b\} \setminus \{y\}$. Therefore, $v_i = 0, i \in \{1, ..., b\} \setminus \{y\}$ and $v_y = x$. Hence, $w = y v_y = yx$.

Thus, in both the cases, $(x, y, w) \in \mathcal{X}$.

For $b = 1$, it is straightforward to verify that $\mathcal{M} = \mathcal{X}$. For $b > 1$, observe that $(\frac{a}{b}, 1, a) \in \mathcal{M} \setminus \mathcal{X}$. □

The set $\mathcal{M}$ is a strong relaxation of $\mathcal{X}$. In particular, the convex hulls of $\mathcal{M}$ and $\mathcal{X}$ are exactly the same and equal to the linear programming (LP) relaxation of $\mathcal{M}$, i.e.

$$\text{conv}(\mathcal{X}) = \text{conv}(\mathcal{M}) = \text{relax}(\mathcal{M}).$$

This follows from earlier work on McCormick envelopes of a bilinear term [5, 70] and observing that $y \in \{0, b\}$ at extreme points of relax($\mathcal{M}$). The remaining question is how

strong the LP relaxations of $\mathcal{U}$ and $\mathcal{B}$ are. Towards this end, we first show that the LP relaxations of $\mathcal{U}$ and $\mathcal{B}$ are generally weaker than that of $\mathcal{M}$.

Proposition 3.2. *The relaxations of the three sets $\mathcal{M}, \mathcal{B}$, and $\mathcal{U}$ compare as follows.*

1. $\text{relax}(\mathcal{M}) \subseteq \text{Proj}_{x,y,w}(\text{relax}(\mathcal{B}))$ *with strict inclusion if and only if $b \neq 2^\gamma - 1$, for any positive integer γ.*

2. $\text{relax}(\mathcal{M}) \subseteq \text{Proj}_{x,y,w}(\text{relax}(\mathcal{U}))$ *and the inclusion is strict if and only if $b \geq 2$.*

Proof. From Proposition 3.1 we have that $\mathcal{X} = \text{Proj}_{x,y,w}(\mathcal{B})$. This implies $\mathcal{X} \subseteq \text{Proj}_{x,y,w}(\text{relax}(\mathcal{B}))$ and since $\text{Proj}_{x,y,w}(\text{relax}(\mathcal{B}))$ is a convex set, we obtain that $\text{conv}(\mathcal{X}) \subseteq \text{Proj}_{x,y,w}(\text{relax}(\mathcal{B}))$. Now, in the above discussion we argued that $\text{relax}(\mathcal{M}) = \text{conv}(\mathcal{X})$ which implies the inclusion $\text{relax}(\mathcal{M}) \subseteq \text{Proj}_{x,y,w}(\text{relax}(\mathcal{B}))$.

Next we verify that the inclusion $\text{relax}(\mathcal{M}) \subseteq \text{Proj}_{x,y,w}(\text{relax}(\mathcal{B}))$ is strict if and only if $b \neq 2^\gamma - 1$, for any $\gamma \in \mathbb{Z}_+$. First suppose that $b \neq 2^\gamma - 1$, for all $\gamma \in \mathbb{Z}_+$. Recall that $k = \lfloor \log_2 b \rfloor + 1$. Take a point (x, y, w, z, v) constructed as follows

$$z_i = \frac{1}{k},\ v_i = az_i, \qquad \forall i \in \{1, \ldots, k\}$$
$$y = \frac{2^k - 1}{k},\ w = ay,\ x = \frac{a}{k}.$$

It is easily verified that this point satisfies the linear constraints of $\text{relax}(\mathcal{B})$. Since for $k \geq 2$ we have that $\frac{2^k-1}{k} < 2^{k-1} \leq b$, the upper bound on y is also satisfied. Hence this point belongs to $\text{relax}(\mathcal{B})$. However, because $b \neq 2^\gamma - 1$ and $k = \lfloor \log_2 b \rfloor + 1$, it follows that $b < 2^k - 1$. Therefore $w > bx$ and the chosen point does not belong to $\text{relax}(\mathcal{M})$.

Now suppose that $b = 2^\gamma - 1$, for some $\gamma \in \mathbb{Z}_{++}$. Since $k = \lfloor \log_2 b \rfloor + 1$, we have that $b = 2^k - 1$. Consider any point $(x, y, w, z, v) \in \text{relax}(\mathcal{B})$. Since $v_i \leq x$ for all $i \in \{1, \ldots, k\}$,

$$\begin{aligned} w = \sum_{i=1}^{k} 2^{i-1} v_i &\leq \sum_{i=1}^{k} 2^{i-1} x \\ &= (2^k - 1)x \\ &= bx. \end{aligned}$$

Similarly, $v_i \leq az_i$ and $v_i \geq x + az_i - a$ for all $i \in \{1, \ldots, k\}$, imply that $w \leq ay$ and $w \geq bx + ay - ab$, respectively. Hence, the point (x, y, w) belongs to $\text{relax}(\mathcal{M})$.

The proof for the inclusion $\text{relax}(\mathcal{M}) \subseteq \text{Proj}_{x,y,w}(\text{relax}(\mathcal{U}))$ is similar to that for $\text{relax}(\mathcal{M}) \subseteq \text{Proj}_{x,y,w}(\text{relax}(\mathcal{B}))$. Observe that for $b = 1$, the two sets $\text{relax}(\mathcal{M})$ and $\text{Proj}_{x,y,w}(\text{relax}(\mathcal{U}))$ are exactly the same because $y = z_1$ and $w = v_1$.

Now suppose that $b \geq 2$ is some positive integer. Construct a point $(x, y, w, z, v) \in \text{relax}(\mathcal{U})$ as follows

$$z_i = \frac{1}{b},\ v_i = az_i, \qquad \forall i \in \{1, \ldots, b\}$$
$$y = \frac{b+1}{2},\ w = ay,\ x = \frac{a}{b}.$$

Thus, $w = \frac{a(b+1)}{2}$. For $b > 1$, it follows that $w > a = bx$ and hence this point does not belong to $\text{relax}(\mathcal{M})$. □

Now we compare the relaxations of $\mathcal{B}$ and $\mathcal{U}$. We first observe that for $b = 2$, the two sets $\mathcal{B}$ and $\mathcal{U}$ are almost the same except that $\mathcal{U}$ has an additional constraint $z_1 + z_2 \leq 1$, thus giving us $\text{relax}(\mathcal{U}) \subset \text{relax}(\mathcal{B})$. Proposition 3.2 implies that if $b = 2^\gamma - 1$ for some integer $\gamma \geq 2$, then $\text{Proj}_{x,y,w}(\text{relax}(\mathcal{B})) = \text{relax}(\mathcal{M}) = \text{conv}(\mathcal{M}) \supset \text{Proj}_{x,y,w}(\text{relax}(\mathcal{U}))$. Hence the relaxation of $\mathcal{B}$ is stronger (in the original (x, y, w)-space) than the relaxation of $\mathcal{U}$. However, this dominance does not always hold true.

Proposition 3.3. *Let $b \geq 3$ be an integer such that $b \neq 2^\gamma - 1$, for any $\gamma \in \mathbb{Z}_{++}$. Then in general,*

1. $\text{Proj}_{x,y,w}(\text{relax}(\mathcal{B})) \setminus \text{Proj}_{x,y,w}(\text{relax}(\mathcal{U})) \neq \emptyset$.
2. $\text{Proj}_{x,y,w}(\text{relax}(\mathcal{U})) \setminus \text{Proj}_{x,y,w}(\text{relax}(\mathcal{B})) \neq \emptyset$.

Proof. Consider the point $(\epsilon/B, b, 0)$ where $B = 2^k - 1$ and $\epsilon \in (0, a(B-b)]$. Since $b \neq 2^\gamma - 1$, for any $\gamma \in \mathbb{Z}_{++}$, and $B = 2^k - 1$, it must be that $b < B$ and hence the choice of ϵ is well defined. We will first show that there exists a $z \in [0, 1]^k$ such that $(\epsilon/B, b, 0, z, 0) \in \text{relax}(\mathcal{B})$. Equivalently, we have to show that there exists a $z \in [0, 1]^k$ such that

$$\sum_{i=1}^{k} 2^{i-1} z_i = b$$
$$0 \leq az_i \leq a - \frac{\epsilon}{B} \qquad \forall i \in \{1, \ldots, k\}.$$

Consider the hypercube $[0, 1 - \epsilon/aB]^k$. Then,

$$\begin{aligned}
\max\Big\{ \sum_{i=1}^{k} 2^{i-1}\zeta_i \colon \zeta \in [0, 1 - \epsilon/aB]^k \Big\} &= \Big(1 - \frac{\epsilon}{aB}\Big) \sum_{i=1}^{k} 2^{i-1} \\
&= \Big(1 - \frac{\epsilon}{aB}\Big) (2^k - 1) \\
&= B - \frac{\epsilon}{a} \\
&\geq b,
\end{aligned}$$

where the last inequality follows from the construction of ϵ. Clearly the minimum of the expression $\sum_{i=1}^{k} 2^{i-1}\zeta_i$ over $[0, 1 - \epsilon/aB]^k$ is 0. Then, by continuity of $\sum_{i=1}^{k} 2^{i-1}\zeta_i$, there must exist some $\hat{z} \in [0, 1 - \epsilon/aB]^k$ such that $\sum_{i=1}^{k} 2^{i-1}\hat{z}_i = b$. Hence the point $(\epsilon/B, b, 0, \hat{z}, 0) \in \text{relax}(\mathcal{B})$ and consequently, $(\epsilon/B, b, 0) \in \text{Proj}_{x,y,w}(\text{relax}(\mathcal{B}))$. To show that $(\epsilon/B, b, 0) \notin \text{Proj}_{x,y,w}(\text{relax}(\mathcal{U}))$, suppose for the sake of contradiction that there exist some $(\bar{z}, \bar{v})$ such that $(\epsilon/B, b, 0, \bar{z}, \bar{v}) \in \text{relax}(\mathcal{U})$. Then, $w = 0$ implies that $\bar{v}_i = 0$, $\forall i = 1, \ldots, b$, and $y = b$ implies that $\bar{z}_b = 1$. On the other hand,

$$\begin{aligned}
x + a\bar{z}_b - a &= \frac{\epsilon}{B} \\
&> 0 \\
&= \bar{v}_b,
\end{aligned}$$

a contradiction to the feasibility of $(\epsilon/B, b, 0, \bar{z}, \bar{v})$.

Finally, we construct a point $(x, y, w) \in \text{Proj}_{x,y,w}(\text{relax}(\mathcal{U})) \setminus \text{Proj}_{x,y,w}(\text{relax}(\mathcal{B}))$. Consider a point in $\text{relax}(\mathcal{U})$ such that

$$\begin{aligned}
\bar{z}_i = \frac{1}{b},\ \bar{v}_i = az_i, \qquad \forall i \in \{1, \ldots, b\} \\
y = \frac{b+1}{2},\ w = ay,\ x = \frac{a}{b}.
\end{aligned}$$

Suppose, for the purpose of contradiction, there exist z and v such that $(x, y, w, z, v) \in \text{relax}(\mathcal{B})$. Then, $w - ay = 0$ implies

$$\sum_{i=1}^{k} 2^{i-1}(v_i - az_i) = 0.$$

Since $v_i \leq az_i, \forall i \in \{1, \ldots, k\}$, it follows from the above equality that $v_i = az_i$ and consequently $v_i = az_i \leq x$, $\forall i \in \{1, \ldots, k\}$. Thus,

$$\begin{aligned} y &= \sum_{i=1}^{k} 2^{i-1} z_i \\ &\leq \frac{\sum_{i=1}^{k} 2^{i-1} x}{a} \\ &\leq \frac{2^k - 1}{2^{k-1}}, \end{aligned}$$

since $x = a/b$ and $b \geq 2^{k-1}$. One can verify that $(2^k - 1)/2^{k-1} < 2$, which leads to $y < 2$. However this is a contradiction because we chose $y = (b+1)/2$ and assumed $b \geq 3$. Hence we have shown that $\text{Proj}_{x,y,w}(\text{relax}(\mathcal{U})) \setminus \text{Proj}_{x,y,w}(\text{relax}(\mathcal{B}))$ is nonempty. □

In the two MILP reformulations $\mathcal{U}$ and $\mathcal{B}$, the number of additional binary variables is b and $\lfloor \log_2 b \rfloor$, respectively. More binary variables for $\mathcal{U}$ implies more number of branchings to be performed in a branch-and-bound algorithm and thus, possibly a higher computational time. Hence, although the strengths of the LP relaxations of $\mathcal{U}$ and $\mathcal{B}$ are incomparable, we do not consider the reformulation $\mathcal{U}$. Our purpose, as detailed in §3.3, is to tighten relax($\mathcal{B}$) using valid inequalities.

3.2.2 Reformulations of (MIBLP)

Suppose that we perform binary expansion of integer variable $y_j, \forall j \in \{1, \ldots, n\}$, in (MIBLP) and use the reformulation $\mathcal{B}$ for each bilinear term. For any given j, we use the same binary expansion variable z^j for all the bilinear variables $w_{lj}, \forall l \in \{1, \ldots, m\}$. This gives us the following extended MILP reformulation,

$$\begin{aligned} \min \quad & \sum_{l=1}^{m} \sum_{j=1}^{n} Q_{0lj} w_{lj} + f_0^\top x + g_0^\top y \\ \text{s.t.} \quad & Ax + Gy \leq h_0 \\ & \sum_{l=1}^{m} \sum_{j=1}^{n} Q_{tlj} w_{lj} + f_t^\top x + g_t^\top y \leq h_t, \qquad t = 1, \ldots, p, \\ & (x_l, y_j, w_{lj}, z^j, v^{lj}) \in \mathcal{B}_{lj}, \quad l = 1, \ldots, m, \ j = 1, \ldots, n. \end{aligned} \tag{B-MIBLP}$$

Alternatively, linearizing every bilinear term in (MIBLP) using the set $\mathcal{M}$ gives us the

following MILP relaxation.

$$\begin{aligned} \min \quad & \sum_{l=1}^{m}\sum_{j=1}^{n} Q_{0lj} w_{lj} + f_0^\top x + g_0^\top y \\ \text{s.t.} \quad & Ax + Gy \le h_0 \\ & \sum_{l=1}^{m}\sum_{j=1}^{n} Q_{tlj} w_{lj} + f_t^\top x + g_t^\top y \le h_t, \qquad t = 1, \dots, p, \\ & (x_l, y_j, w_{lj}) \in \mathcal{M}_{lj}, \quad l = 1, \dots, m,\ j = 1, \dots, n. \end{aligned} \tag{M-MIBLP}$$

On comparing the above two formulations, we note that (M-MIBLP) has at most $4mn$ more constraints than (MIBLP) whereas (B-MIBLP) has at most $(m+1)\sum_{j=1}^{n} k_j$ more variables and $4m\sum_{j=1}^{n} k_j$ more constraints than (MIBLP), where $k_j = \lfloor \log_2 u_j \rfloor + 1$ for $j = 1, \dots, n$.

Let $\eta^*(\cdot)$ denote the optimum value of a problem. Since $\mathcal{X} = \text{Proj}_{x,y,w}(\mathcal{B})$, it follows that solving (B-MIBLP) gives us the true optimal value of (MIBLP). On the contrary, because $\mathcal{X}$ is a strict subset of $\mathcal{M}$ for $b \ge 2$, (M-MIBLP) is a relaxation of (MIBLP). Thus, we have

$$\eta^*(\text{M-MIBLP}) \le \eta^*(\text{MIBLP}) = \eta^*(\text{B-MIBLP}).$$

3.3 Facets of $\text{conv}(\mathcal{B})$

In this section, the focus is on solving reformulation (B-MIBLP). We conduct a polyhedral study of $\text{conv}(\mathcal{B})$ and describe $\text{conv}(\mathcal{X})$ in the (x, y, w, z, v)-space. The aim is to use these facets as valid inequalities in a branch-and-cut algorithm for solving problem (B-MIBLP).

We first provide some definitions that will be used in this section. Let

$$\mathcal{K} := \left\{ z \in \{0,1\}^k \colon \sum_{i=1}^{k} 2^{i-1} z_i \le b \right\} \tag{56}$$

be a $\{0,1\}$-knapsack set and let

$$\mathcal{R}^{\mathcal{K}} := \Big\{ (x, z, v) \in \Re_+ \times \Re^k \times \Re^k \colon z \in \mathcal{K}, x \le a, v_i = x z_i, \forall i \in \{1, \dots, k\} \Big\}. \tag{57}$$

Note that since $\mathcal{K} \subseteq \{0,1\}^k$, the McCormick linearization of $v_i = x z_i$ is exact for all i, and

hence $\mathcal{R}^{\mathcal{K}}$ can be rewritten as

$$\mathcal{R}^{\mathcal{K}} = \Big\{(x,z,v) \in \quad \Re_+ \times \Re^k \times \Re^k \colon z \in \mathcal{K}, \\ v_i \geq 0, v_i \leq az_i, v_i \leq x, v_i \geq x + az_i - a, \forall i \in \{1,\ldots,k\}\Big\}.$$

From the definition of $\mathcal{B}$ in (55), it follows that the variables y and w are just linear functions of z and v, respectively. Hence

$$\text{conv}(\mathcal{B}) = \left\{(x,y,w,z,v) \colon y = \sum_{i=1}^{k} 2^{i-1} z_i,\ w = \sum_{i=1}^{k} 2^{i-1} v_i,\ (x,z,v) \in \text{conv}(\mathcal{R}^{\mathcal{K}})\right\}. \quad (58)$$

Since $\text{conv}(\mathcal{X}) = \text{Proj}_{x,y,w}(\text{conv}(\mathcal{B}))$, equation (58) tells us that an extended representation of $\text{conv}(\mathcal{X})$ can be easily obtained once we know $\text{conv}(\mathcal{R}^{\mathcal{K}})$. Towards this end, we first state some general results on sets defined by products of variables.

3.3.1 Convex hulls of unconstrained bilinear terms

In this subsection, we henceforth denote χ and ρ to be vectors of variables and ω to be a conformable matrix of variables obtained as $\omega = \chi\rho^\top$. Let $\mathcal{X}^+$ be a general bilinear set defined as follows

$$\mathcal{X}^+ := \left\{(\chi,\rho,\omega) \in \Re^N_+ \times \Re^M \times \Re^{N\times M} \colon \omega = \chi\rho^\top, \chi \in \Theta, \rho \in \Upsilon\right\}, \quad (59)$$

where $\chi = (\chi^{(1)},\ldots,\chi^{(m)})$ such that $\chi^{(i)} \in \Re^{n_i}_+$ for all $i = 1,\ldots,m$, and $N = \sum_{i=1}^m n_i$. Also, $\omega = (\omega^{(1)},\ldots,\omega^{(m)})$ where $\omega^{(i)} \in \Re^{n_i \times M}$, and Υ is some (possibly mixed integer) subset of $\Re^M$ such that the convex hull of Υ is a polyhedron. Before defining Θ, we reiterate an old definition.

Definition 3.1 (Tawarmalani et al. [105])**.** The set Θ is said to be *orthogonal disjunctive* if there exist subsets $\Theta^{(i)} \subseteq \Theta$ that satisfy the following two conditions,

1. $\chi \in \Theta^{(i)}$ implies that $\chi^{(j)} = \mathbf{0}$, for all $j \neq i$, and
2. $\chi \in \Theta$ implies that there exist points ${}^{(i)}\tau \in \text{conv}(\Theta^{(i)})$, for all $i \in I \subseteq \{1,\ldots,m\}$, such that $\chi \in \text{conv}(\cup_{i\in I}\, {}^{(i)}\tau)$.

The first condition imposes orthogonality on elements of Θ because if ${}^{(i)}\tau \in \Theta^{(i)}$ and ${}^{(j)}\tau \in \Theta^{(j)}$, then ${}^{(i)}\tau \perp {}^{(j)}\tau$. The second condition allows a disjunctive representation for

the convex hull of Θ in the sense that $\text{conv}(\Theta) = \text{conv}(\bigcup_{i=1}^{m} \Theta^{(i)})$ (cf. Tawarmalani et al. [105], Claim 1 of Theorem 2.1). It is sometimes nontrivial to verify this second condition. A prime example when it holds true is if $\Theta = \cup_{i=1}^{m} \Theta^{(i)}$. However it is not necessary for Θ to be expressed as a union of $\Theta^{(i)}$.

We further expand on the above definition by enforcing special structure on the subsets $\Theta^{(i)}$.

Definition 3.2. Θ is said to be *simplicial* orthogonal disjunctive if Θ is orthogonal disjunctive and for all $i = 1, \ldots, m$, we have

$$\Theta^{(i)} = \left\{ \chi \geq \mathbf{0} \colon {\sigma^{(i)}}^{\top} \chi^{(i)} \leq \sigma_0^{(i)},\ \chi^{(j)} = \mathbf{0}, \forall j \neq i \right\}, \tag{60}$$

to be a low-dimensional simplex in $\Re^N$ for some $\sigma^{(i)} \in \Re^{n_i}_{++}$ and $\sigma_0^{(i)} > 0$.

We note that although Θ is assumed to be orthogonal disjunctive, the set $\mathcal{X}^+$ is not orthogonal disjunctive, particularly since there is a common variable ρ. Hence the convex hull of $\mathcal{X}^+$ does not follow directly from Tawarmalani et al. [105], Theorem 2.1. This leads us to the main result of this subsection.

Theorem 3.1. *Let* $\text{conv}(\Upsilon) = \{\rho \colon \Pi\rho \leq \pi_0\}$ *be a nonempty polyhedron and* Θ *be a simplicial orthogonal disjunctive set as per Definition 3.2. Denote* $\omega_{t\cdot}^{(i)}$ *as the* t^{th} *row of the submatrix* $\omega^{(i)}$*. Then, the closure convex hull of* $\mathcal{X}^+$ *is given by*

$$\begin{aligned} \text{cl}\,\text{conv}(\mathcal{X}^+) = \Bigg\{ (\chi, \rho, \omega) \colon & \Pi\, {\omega_{t\cdot}^{(i)}}^{\top} \leq \pi_0 \chi_t^{(i)}, \quad i = 1, \ldots, m, t = 1, \ldots, n_i \\ & \Pi\rho - \sum_{i=1}^{m} \frac{\Pi {\sigma^{(i)}}^{\top} \omega^{(i)}}{\sigma_0^{(i)}} \leq \pi_0 \left[1 - \sum_{i=1}^{m} \frac{{\sigma^{(i)}}^{\top} \chi^{(i)}}{\sigma_0^{(i)}} \right] \\ & \sum_{i=1}^{m} \frac{{\sigma^{(i)}}^{\top} \chi^{(i)}}{\sigma_0^{(i)}} \leq 1,\ \chi \geq \mathbf{0} \Bigg\}. \end{aligned}$$

Proof. The following two claims are straightforward to verify.

Claim 1 : $\text{conv}(\mathcal{X}^+) = \text{conv}\{(\chi, \rho, \omega) \colon \omega = \chi\rho^{\top}, \chi \in \Theta, \rho \in \text{conv}(\Upsilon)\}$. The forward inclusion ($\subseteq$) is trivial since $\Upsilon \subseteq \text{conv}(\Upsilon)$, whereas the reverse inclusion ($\supseteq$) is by Caratheodory's theorem applied to $\text{conv}(\Upsilon)$.

Claim 2 : If $(\chi, \rho, \omega) \in \text{ext}\,\text{conv}(\mathcal{X}^+)$, then $\chi \in \text{ext}\,\Theta$. Furthermore, since Θ is simplicial orthogonal disjunctive, it must be that $\chi \in \text{ext}\,\Theta^{(i)}$, for some $i \in \{1, \ldots, m\}$.

The set of extreme points of the simplex $\Theta^{(i)}$ is

$$\operatorname{ext}\Theta^{(i)} = \{(\mathbf{0},\ldots,\mathbf{0})\} \cup \bigcup_{t=1}^{n_i}\left(\mathbf{0},\ldots,\frac{\sigma_0^{(i)}\mathbf{e}_t}{\sigma_t^{(i)}},\mathbf{0},\ldots,\mathbf{0}\right).$$

Define polyhedron $\Psi^{(i,t)}$, for all $i = 1,\ldots,m, t = 1,\ldots,n_i$, as

$$\Psi^{(i,t)} = \left\{(\chi,\rho,\omega)\colon \chi^{(j)} = \mathbf{0}, \forall j \neq i, \chi^{(i)} = \frac{\sigma_0^{(i)}\mathbf{e}_t}{\sigma_t^{(i)}},\ \Pi\rho \leq \pi_0,\ \omega^{(j)} = \mathbf{0}, \forall j \neq i, \omega^{(i)} = \frac{\sigma_0^{(i)}\mathbf{e}_t}{\sigma_t^{(i)}}\rho^\top\right\},$$

and $\Psi^{(0)} = \{(\chi,\rho,\omega)\colon \chi = \mathbf{0},\ \Pi\rho \leq \pi_0,\ \omega = \mathbf{0}\}$.

Take a point $\chi \in \Theta$. Since Θ is orthogonal disjunctive by assumption and $\Theta^{(i)}$ is convex, there exist points ${}^{(i)}\tau \in \Theta^{(i)}, i \in I \subseteq \{1,\ldots,m\}$, such that $\chi \in \operatorname{conv}(\cup_{i\in I}\,{}^{(i)}\tau)$. Since $\Theta^{(i)}$ is a simplex, clearly, the following is true for ${}^{(i)}\tau^{(i)}$, for all i.

$${}^{(i)}\tau^{(i)} = \sum_{t=1}^{n_i} {}^{(i)}\tau_t^{(i)}\mathbf{e}_t = \sum_{t=1}^{n_i} \frac{{}^{(i)}\tau_t^{(i)}\sigma_t^{(i)}}{\sigma_0^{(i)}}\,\frac{\sigma_0^{(i)}\mathbf{e}_t}{\sigma_t^{(i)}}.$$

Hence any point $(\chi,\rho,\omega) \in \mathcal{X}^+$ can be written as a convex combination of points in $\Psi^{(0)}, \Psi^{(i,t)}, \forall i,t$, and it must be that

$$\operatorname{conv}(\mathcal{X}^+) = \operatorname{conv}\left(\Psi^{(0)} \cup \bigcup_{i=1}^{m}\bigcup_{t=1}^{n_i}\Psi^{(i,t)}\right).$$

Then Balas [16], Theorem 2.1 implies the following extended formulation for $\operatorname{cl}\operatorname{conv}(\mathcal{X}^+)$.

$$\begin{aligned}\operatorname{cl}\operatorname{conv}(\mathcal{X}^+) = \operatorname{Proj}_{\chi,\rho,\omega}\Big\{\big(\{{}^{(i,t)}\chi,\,{}^{(i,t)}\rho,{}^{(i,t)}\omega\}_{\substack{i\in[m]\\ t\in[n_i]}},\chi,\rho,\omega,\lambda\big)\colon \quad & \lambda \geq \mathbf{0},\ \lambda_0 + \sum_{i,t}\lambda_{it} = 1\\ & \Pi\,{}^{(i,t)}\rho \geq \pi_0\lambda_{it},\ \Pi\,{}^{(0)}\rho \geq \pi_0\lambda_0\\ & {}^{(i,t)}\chi^{(i)} = \frac{\sigma_0^{(i)}\lambda_{it}\mathbf{e}_t}{\sigma_t^{(i)}},\ {}^{(i,t)}\chi^{(j)} = \mathbf{0}\ \forall i,t,j \neq i\\ & {}^{(i,t)}\omega^{(i)} = \frac{\sigma_0^{(i)}\mathbf{e}_t}{\sigma_t^{(i)}}\,{}^{(i,t)}\rho^\top,\ {}^{(i,t)}\omega^{(i)} = \mathbf{0}\ \forall i,t,j \neq i\\ & \chi = \sum_{i,t}{}^{(i,t)}\chi,\ \rho = {}^{(0)}\rho + \sum_{i,t}{}^{(i,t)}\rho,\ \omega = \sum_{i,t}{}^{(i,t)}\omega\Big\},\end{aligned}$$

where ${}^{(i,t)}\chi$ denotes the copy of χ in the $(i,t)^{th}$ disjunction. Similarly for ρ and ω. In order to obtain the projection, first note that $\chi_t^{(i)} = {}^{(i,t)}\chi_t^{(i)}$. Hence, $\lambda_{it} = \sigma_t^{(i)}\chi_t^{(i)}/\sigma_0^{(i)}$. The convex combination condition $\sum_{i,t}\lambda_{it} \leq 1$ then implies the projected inequality

$$\sum_{i=1}^{m}\frac{{\sigma^{(i)}}^\top\chi^{(i)}}{\sigma_0^{(i)}} \leq 1.$$

Now, $\omega_{t\cdot}^{(i)} = \sum_t {}^{(i,t)}\omega_{t\cdot}^{(i)} = \sigma_0^{(i)}\ {}^{(i,t)}\rho^\top/\sigma_t^{(i)}$ and hence

$$ {}^{(i,t)}\rho^\top \quad = \quad \frac{\sigma_t^{(i)}\ {}^{(i,t)}\omega_{t\cdot}^{(i)}{}^\top}{\sigma_0^{(i)}}. $$

Substituting the above in $\Pi\,{}^{(i,t)}\rho \geq \pi_0\lambda_{it}$ and using $\lambda_{it} = \sigma_t^{(i)}\chi_t^{(i)}/\sigma_0^{(i)}$ gives the following upon cancellation of terms,

$$ \Pi\,\omega_{t\cdot}^{(i)}{}^\top \leq \pi_0\chi_t^{(i)}. $$

Finally, it remains to project out ${}^{(0)}\rho$ and λ_0. The following two identities

$$ \begin{aligned} \lambda_0 = 1 - \sum_{i,t}\lambda_{it} \quad &= \quad 1 - \sum_{i=1}^m \frac{\sigma^{(i)}{}^\top\chi^{(i)}}{\sigma_0^{(i)}} \\ {}^{(0)}\rho \quad &= \quad \rho - \sum_{i,t}\ {}^{(i,t)}\rho \end{aligned} $$

imply the last inequality

$$ \Pi\rho - \sum_{i=1}^m \frac{\Pi\sigma^{(i)}{}^\top\omega^{(i)}}{\sigma_0^{(i)}} \leq \pi_0\left[1 - \sum_{i=1}^m \frac{\sigma^{(i)}{}^\top\chi^{(i)}}{\sigma_0^{(i)}}\right]. $$

□

A closer look at the statement of Theorem 3.1 reveals that the proposed convex hull defining inequalities can be obtained by multiplying each defining inequality in the polyhedral description of $\text{conv}(\Upsilon)$ by the variable bound factors $\chi_t^{(i)}$, for all $i = 1,\ldots,m, t = 1,\ldots,n_i$, and the factor $1 - \sum_{i=1}^m \sigma^{(i)}{}^\top\chi^{(i)}/\sigma_0^{(i)}$. Such variable bound factor multiplication forms the basic principle of the Reformulation Linearization Technique (RLT) proposed by Sherali and Adams [97]. Since the variable factors for $\text{conv}(\mathcal{X}^+)$ are linear in χ, we say that $\text{conv}(\mathcal{X}^+)$ is a rank-1 RLT.

Corollary 3.1. *The convex hull of $\mathcal{X}^+$ has RLT rank equal to one.*

In Theorem 3.1, we assumed that the projection of $\Theta^{(i)}$ onto $\Re_+^{n_i}$ is described by a bounded halfspace in the positive orthant. However, one can easily generalize the forgoing result under the assumption that this projection is described by any arbitrary simplex. The only property we really need is that this projection be a simplex, i.e. have at most $n_i + 1$ affinely independent extreme points, so that every point $\chi^{(i)} \in \text{Proj}_{\chi^{(i)}}\Theta^{(i)}$ is uniquely characterized by the extreme points of this projection.

We next present an immediate implication of Theorem 3.1 that is also useful in §1.5.2.1.

Corollary 3.2. *Let* $\Delta = \{\chi \geq \mathbf{0}\colon \sum_{t=1}^{N} \chi_t = 1\}$ *be the standard* N*-dimensional simplex in* $\Re_+^N$.

$$\operatorname{conv}\Big(\{(\chi,\rho,\omega)\colon \chi\in\Delta, \Pi\rho\leq\pi_0, \omega=\chi\rho^\top\}\Big) \;=\; \Big\{(\chi,\rho,\omega)\colon \quad \Pi\,\omega_{t\cdot}^\top \leq \pi_0\chi_t, \quad t=1,\ldots,N \\ \textstyle\sum_{t=1}^{N}\omega_{t\cdot}^\top = \rho,\ \chi\in\Delta\Big\}.$$

Proof. Set $m = 1, n_1 = N$, and $\Theta = \Delta$ in Theorem 3.1. Using similar steps to address the equality constrained simplex, we obtain the projection onto the original space. □

Remark 3.1 (Connections to RLT). Now consider a special case of Theorem 3.1 with $m = 1, n_i = N, \Theta = \Delta$, as in Corollary 3.2. We further assume that $\operatorname{conv}(\Upsilon)$ is a polytope. In the proof of Theorem 3.1, we used the identity $\operatorname{conv}(\mathcal{X}^+) = \operatorname{conv}(\cup_{i,t}\Psi^{(i,t)})$. Thus we can rewrite

$$\operatorname{conv}(\mathcal{X}^+) = \operatorname{conv}\{(\chi,\rho,\omega)\colon \omega=\chi\rho^\top, \chi\in\Delta, \chi\in\{0,1\}^N, \rho\in\operatorname{conv}(\Upsilon)\}.$$

Since $\operatorname{conv}(\Upsilon)$ is assumed to be a polytope, there exist finite lower and upper bounds on ρ_j for all $j = 1,\ldots,M$. Also, note that $\chi \in \{0,1\}^N$. Then, we can exactly reformulate the bilinear term $\omega_{ij} = \chi_i\rho_j$ using its McCormick envelopes (13), for all i, j. This implies that

$$\operatorname{conv}(\mathcal{X}^+) = \operatorname{conv}\{(\chi,\rho,\omega)\colon \mathfrak{G}\chi + \mathfrak{B}\rho + \mathfrak{C}\omega \geq b \\ \chi\in\Delta, \chi\in\{0,1\}^N\}$$

for some matrices $(\mathfrak{G}, \mathfrak{B}, \mathfrak{C})$ and vector b. Observe that the set on the right hand side is the convex hull of a mixed integer linear set where the binary variables χ are SOS1. The result of Sherali et al. [99] implies that $\operatorname{conv}(\mathcal{X}^+)$ can be obtained via a RLT procedure that involves multiplying each linear constraint from the system $\mathfrak{G}\chi + \mathfrak{B}\rho + \mathfrak{C}\omega \geq b$ by χ_i, for $i = 1,\ldots,N$, and $1 - \sum_{i=1}^{N}\chi_i$. Sherali et al. show that $\chi_i\chi_j = 0, \forall i, j$ and $\chi_i \in \{0,1\}$ strenghtens $\chi_i^2 = \chi_i$. After carrying out the multiplication on the McCormick envelopes, we obtain $\omega_{ij}\chi_k = 0, \forall i, j, k$. On substituting $\omega_{ij} = \chi_i\rho_j$ we get exactly the linear description from the statement of Corollary 3.2. We used the result on disjunctive programming of Balas [16] in our proof. Our derivation is stronger since it allows two generalizations:

1) Θ is a orthogonal disjunction of simplices, and 2) $\text{conv}(\Upsilon)$ is a polyhedron and hence it may not be possible to use variable bounds to reformulate $\mathcal{X}^+$ as a mixed integer linear set.

Assuming that $\Theta = \Delta^{\leq} = \{\chi \geq \mathbf{0}\colon \sum_{t=1}^{N} \chi_t \leq 1\}$ and $\text{conv}(\Upsilon)$ is a polytope, we next state another result that gives the convex hull of the intersection of general bilinear sets that share a common variable bounded within a simplex. This result is essentially a set analogy of Rikun [89], Theorem 1.4 on convex envelope of summation of functions.

Proposition 3.4. *Define* $\mathcal{X}^{\infty} := \cap_{r\in R}\mathcal{X}_r^+$*, where for any* $r \in R$*,*

$$\mathcal{X}_r^+ := \{(\chi, \rho^r, \omega^r)\colon \chi \in \Delta^{\leq}, \rho^r \in \Upsilon_r, \omega^r = \chi(\rho^r)^\top\}$$

and $\text{conv}(\Upsilon_r)$ *is a polytope. Then,* $\text{conv}(\mathcal{X}^{\infty}) = \cap_{r\in R}\,\text{conv}(\mathcal{X}_r^+)$.

Proof. Consider optimizing any arbitrary linear function over $\mathcal{X}^{\infty}$.

$$\begin{array}{rcllcll} \theta^* & = & \min & \sum_r G_r \bullet \omega^r + \sum_r d_m^\top \rho^r + c^\top\chi & = & \min & \sum_r G_r \bullet \omega^r + \sum_r d_r^\top \rho^r + c^\top\chi \\ & & \text{s.t.} & (\chi, \{\rho^r\}_r, \{\omega^r\}_r) \in \mathcal{X}^{\infty} & & \text{s.t.} & (\chi, \{\rho^r\}_r, \{\omega^r\}_r) \in \text{conv}(\mathcal{X}^{\infty}) \end{array}$$

where $A \bullet B$ is a Frobenius inner product between two matrices A and B of conformable dimensions. It suffices to show that

$$\begin{array}{rcll} \theta^* & = & \min & \sum_r G_r \bullet \omega^r + \sum_r d_r^\top \rho^r + c^\top\chi \\ & & \text{s.t.} & (\chi, \rho^r, \omega^r) \in \text{conv}(\mathcal{X}_r^+), \qquad r \in R. \end{array}$$

By definition of $\mathcal{X}^{\infty}$, $\text{conv}\,\mathcal{X}^{\infty} \subseteq \cap_r \text{conv}\,\mathcal{X}_r^+$ and hence the $\geq$ inequality is obvious. Now, since $\omega^r = \chi(\rho^r)^\top$ for any point $(\chi, \{\rho^r\}, \{\omega^r\}_r) \in \mathcal{X}^{\infty}$, we rewrite θ^* as

$$\begin{array}{rcllcll} \theta^* & = & \min & \gamma + \sum_r d_r^\top \rho^r + c^\top\chi & = & \min & \gamma + \sum_r d_r^\top \rho^r + c^\top\chi \\ & & \text{s.t.} & \sum_r f_r(\chi, \rho^r) \leq \gamma & & \text{s.t.} & (\chi, \{\rho^r\}, \gamma) \in \text{epi}\,\text{cvx} \sum_r f_r(\chi, \rho^r) \\ & & & \chi \in \Delta^{\leq},\ \rho^r \in \Upsilon_r, \quad r \in R, & & & \end{array}$$

where $f_r(\chi, \rho^r) = \chi^\top G_r \rho^r$ and $\text{epi}\,\text{cvx}\sum_r f_r(\cdot,\cdot)$ denotes the epigraph of the convex envelope of $\sum_r f_r(\cdot,\cdot)$ over $\Delta^{\leq} \times \prod_r \text{conv}(\Upsilon_r)$. Since $\Delta^{\leq}$ is a simplex and $\text{conv}(\Upsilon_r)$ is a polytope, Rikun's formula [89, Theorem 1.4] for the convex envelope of summation of functions implies

$$\text{cvx}\sum_r f_r(\chi, \rho^r) = \sum_r \text{cvx}\, f_r(\chi, \rho^r).$$

Hence, it follows that

$$\begin{array}{rcll} \theta^* & = & \min & \gamma + \sum_t d_t^\top \rho^t + c^\top \chi \\ & & \text{s.t.} & \sum_t \gamma_t \leq \gamma \\ & & & (\chi, \rho^t, \gamma_t) \in \operatorname{epi} \operatorname{cvx} f_t(\chi, \rho^r), \quad r \in R. \end{array}$$

Define $F_r = \{(\chi, \rho^r, \gamma_r) \colon f_r(\chi, \rho^r) = \gamma_r, \chi \in \Delta^{\leq}, \rho^r \in \Upsilon_r\}$. By definition,

$$F_r = \operatorname{Proj}_{\chi, \rho^r, \omega^r} \{(\chi, \rho^r, \gamma_r, \omega^r) \colon \gamma_r = G_r \bullet \omega^r, (\chi, \rho^r, \omega^r) \in \mathcal{X}_r^+\}.$$

Hence, the convex hull of F_r is

$$\operatorname{conv}(F_r) = \operatorname{Proj}_{\chi, \rho^r, \omega^r} \{(\chi, \rho^r, \gamma_r, \omega^r) \colon \gamma_r = G_r \bullet \omega^r, (\chi, \rho^r, \omega^r) \in \operatorname{conv}(\mathcal{X}_r^+)\}.$$

Also, since $F_r \subseteq \operatorname{epi} f_r$, then $\operatorname{conv}(F_r) \subseteq \operatorname{conv} \operatorname{epi} f_r = \operatorname{epi} \operatorname{cvx} f_r$. Substituting this inclusion and the identity for $\operatorname{conv}(F_r)$ into θ^* we get

$$\begin{array}{rcll} \theta^* & \leq & \min & \gamma + \sum_t d_t^\top \rho^t + c^\top \chi \\ & & \text{s.t.} & \sum_r G_r \bullet \omega^r \leq \gamma \\ & & & (\chi, \rho^r, \omega^r) \in \operatorname{conv}(\mathcal{X}_r^+), \quad r \in R. \end{array} \qquad \begin{array}{rcll} & = & \min & \sum_r G_r \bullet \omega^r + \sum_t d_t^\top \rho^t + c^\top \chi \\ & & \text{s.t.} & (\chi, \rho^r, \omega^r) \in \operatorname{conv}(\mathcal{X}_r^+), \quad r \in R. \end{array}$$

□

We now use the preceding results in our context, where the simplex is a bounded interval on the real line.

Proposition 3.5. *Let $\Upsilon \subset \Re^k$ be some subset such that its convex hull $\operatorname{conv}(\Upsilon) = \{z \colon \Pi z \leq \pi_0\}$ is a polytope for some matrix Π and right hand side π_0. Define $\mathcal{R}^\Upsilon$ as*

$$\mathcal{R}^\Upsilon := \left\{(x, z, v) \in \Re_+ \times \Re^k \times \Re^k \colon z \in \Upsilon, x \leq a, v_i = x z_i, \forall i \in \{1, \ldots, k\}\right\},$$

Then, the convex hull of $\mathcal{R}^\Upsilon$ is a polyhedron given by

$$\begin{aligned} \operatorname{conv}(\mathcal{R}^\Upsilon) = \Big\{(x, z, v) \colon x \in [0, a], \qquad & \Pi v \ - \ \pi_0 x \ \leq \ 0, \\ & \Pi z - \frac{1}{a} \Pi v \ + \ \frac{1}{a} \pi_0 x \ \leq \ \pi_0 \Big\}. \end{aligned} \tag{61}$$

Proof. This result follows from Theorem 3.1 by considering $\chi = x, \rho = z, \omega = v$, and $\Theta = [0, a]$. Since $\operatorname{conv}(\Upsilon)$ is bounded, $\operatorname{conv}(\mathcal{R}^\Upsilon)$ is closed. □

3.3.2 Minimal covers of knapsack

Using equation (58), we obtain that $\text{conv}(\mathcal{B})$ is given by $\text{conv}(\mathcal{R}^{\mathcal{K}})$ and two linear equalities. Proposition 3.5 helps us obtain $\text{conv}(\mathcal{R}^{\mathcal{K}})$ by multiplying the linear inequalities describing $\text{conv}(\mathcal{K})$ with x and $a - x$. It remains to find the convex hull of $\mathcal{K}$.

A complete description of the convex hull of a knapsack set with arbitrary weights is unknown. However, note that $\mathcal{K}$ is a special case of the sequential knapsack polytope studied by Pochet and Weismantel [83]. For a sequential knapsack polytope with arbitrary upper bounds on variables and divisible coefficients (that are not just powers of some natural number), a constructive procedure for obtaining all its exponentially many facets was given by Pochet and Weismantel. The set $\mathcal{K}$ is a special case since the weight of each item in knapsack is a successively increasing power of two. We discuss its properties next.

Consider $\mathcal{K}$ and note that $k = \lfloor \log_2 b \rfloor + 1$. Hence, $2^{k-1} \leq b < 2^k$. Now let the binary expansion of b be given by

$$b = 2^{i_1 - 1} + 2^{i_2 - 1} + \cdots + 2^{i_r - 1} + 2^{k-1}, \tag{62}$$

for some $r \geq 0$. Since $2^{k-1} \leq b < 2^k$, we can assume w.l.o.g that the last exponent in the above equation is $k-1$. Note therefore that the convex hull of $\mathcal{K}$ is full dimensional. We use $r = 0$ to denote that $b = 2^{k-1}$. Let $N := \{1, 2, \ldots, k\}$ and define a function $\sigma \colon 2^N \mapsto \Re_+$ as follows

$$\sigma(C) = \begin{cases} 0 & C = \emptyset, \\ \sum_{i \in C} 2^{i-1} & \text{otherwise.} \end{cases} \tag{63}$$

The function $\sigma(\cdot)$ is monotone in the sense that $\sigma(C_1) \leq \sigma(C_2)$ for any $C_1 \subseteq C_2 \subseteq N$. A key observation is the following.

Observation 3.1.

$$\sigma(C) < \sigma(i^*), \quad \textit{for any } C \subseteq N \textit{ and } i^* > \max\{i \colon i \in C\}. \tag{64}$$

Definition 3.3. A sequence of positive reals $\{a_1, a_2, \ldots\}$ is said to be (weakly) superincreasing if it satisfies $\sum_{\tau=1}^{q} a_\tau < (\leq)\ a_{q+1}$, for $q \geq 1$.

It follows from (64) that the coefficients of $\mathcal{K}$ form a superincreasing sequence. A result from Seymour [95] was used in Laurent and Sassano [65] to construct all the nontrivial facet-defining inequalities for a knapsack with weakly superincreasing weights. We first recall the definition of minimal covers of an arbitrary knapsack and then state the result of Laurent and Sassano.

Definition 3.4. For a knapsack $\tilde{\mathcal{K}} := \{\tilde{z} \in \{0,1\}^n \colon \sum_{i=1}^n \tilde{a}_i \tilde{z}_i \leq \tilde{b}\}$, a subset $\tilde{C} \subseteq \{1,\ldots,n\}$ is called a minimal cover if $\sum_{i\in\tilde{C}} \tilde{a}_i > \tilde{b}$ and $\sum_{i\in\tilde{C}\setminus t} \tilde{a}_i \leq \tilde{b}$ for any $t \in \tilde{C}$.

Proposition 3.6. *(Laurent and Sassano [65], Theorem 2.5 and Corollary 2.6)*

Consider a $\{0,1\}$ *knapsack* $\tilde{\mathcal{K}} := \{\tilde{z} \in \{0,1\}^n \colon \sum_{i=1}^n \tilde{a}_i \tilde{z}_i \leq \tilde{b}\}$ *such that* $\{\tilde{a}_n,\ldots,\tilde{a}_1\}$ *is weakly superincreasing. Define the integers* $\tau_1,\ldots,\tau_q$*, for some* $q \geq 1$*, as*

$$\tau_i = \min\left\{ h > \tau_{i-1} \colon \sum_{j=1}^{i-1} \tilde{a}_{\tau_j} + \tilde{a}_h \leq b \right\}, \quad \forall 1 \leq i \leq q,$$

with $\tau_1 = 1$ *and the intervals* $\mathcal{A}_i := \{\tau_i + 1,\ldots,\tau_{i+1} - 1\}, 1 \leq i \leq q$*. Then,*

1. *The minimal covers of* $\mathcal{K}$ *are the sets*

$$\mathcal{C}_{i,j} = \{\tau_1,\ldots,\tau_i,j\}, \quad j \in \mathcal{A}_i, 1 \leq i \leq q.$$

2. *The nontrivial facets of* $\mathcal{K}$ *are given by the minimal covering inequalities*

$$\tilde{z}_{\tau_1} + \cdots + \tilde{z}_{\tau_i} + \tilde{z}_j \leq i, \quad j \in \mathcal{A}_i, 1 \leq i \leq q.$$

□

Let $\mathcal{C}$ denote the set of minimal covers of $\mathcal{K}$. We provide an explicit description of $\mathcal{C}$ to establish its dependence on the binary expansion of the right hand side b.

Proposition 3.7. *Define* $I := \{i_1,\ldots,i_r,k\}$*, where* b *is given by* (62) *and* $\sigma(I) = b$*. Assume w.l.o.g. that* $i_1 < i_2 < \cdots < i_r < k$*. For any* $j \in N \setminus I$*, let* $I_j := \{i \in I \colon i > j\}$*. Then,*

$$\mathcal{C} = \bigcup_{j\in N\setminus I} \{j, I_j\}. \tag{65}$$

Proof. Note that if $b = 2^k - 1$, then the knapsack inequality in (56) is redundant and the set of covers is empty. Henceforth, assume that $b < 2^k - 1$.

We first verify that elements of the form $\{j, I_j\}$ define a minimal cover. Choose a $j \in N \setminus I$ and let $C = \{j, I_j\}$. Then, $\sigma(I) = \sigma(I \setminus I_j) + \sigma(I_j) = \sigma(I \setminus I_j) + \sigma(C) - 2^{j-1}$. Using (64), we have that $\sigma(I \setminus I_j) < 2^{j-1}$. Hence, $b = \sigma(I) < \sigma(C)$ and C is a valid cover for the knapsack. Since $2^{j-1} < 2^{i-1}$, for $i \in I_j$, we have that for any $q \in C \setminus j$, $\sigma(C \setminus q) < \sigma(I_j) \leq \sigma(I) = b$. Finally, $\sigma(C \setminus j) \leq \sigma(I) \leq b$. Hence, C is a minimal cover.

Now, let $C \in \mathcal{C}$ be a minimal cover of the knapsack. Since I is not a cover by definition, we must have $|C \setminus I| \geq 1$. Define $c_1 := \max\{j \colon j \in C \setminus I\}$ and $T_1 := \{j \in C \colon j > c_1\}$.

Claim 1: $T_1 = I_{c_1}$. By definition of c_1 and T_1, we obtain that $T_1 \subseteq I_{c_1}$. Now take $j \in I_{c_1}$ and suppose for the sake of contradiction that $j \notin C$. Define $c_q := \max\{i \in C \colon i < j\}$. Then $\{c_{q+1}, \ldots, k\} \subseteq C \cap I$ and $b = \sigma(I) \geq \sigma(j) + \sigma(\{c_{q+1}, \ldots, k\}) > \sigma(C)$. Hence C is not a cover, a contradiction. This implies $I_{c_1} \subseteq T_1$ and thus proves our claim.

By the above claim it follows that $\{c_1, I_{c_1}\} \subseteq C$. Since $\{c_1, I_{c_1}\}$ is a cover, by minimality of C, we obtain that $C = \{c_1, I_{c_1}\}$. □

Example 3.1. Let $b = 38 = 2^{2-1} + 2^{3-1} + 2^{6-1}$. Hence, $I = \{2, 3, 6\}$. Then, the set of minimal covers is $\{(1, 2, 3, 6), (4, 6), (5, 6)\}$. □

Proposition 3.6 implies that minimal cover inequalities $z_j + \sum_{i \in I_j} z_i \leq |I_j|$, for $j \notin I$, define all the nontrivial facets of $\text{conv}(\mathcal{K})$. For the sake of completeness, we present direct self-contained proofs for this result.

Direct proof of $\text{conv}(\mathcal{K})$**.**

Proposition 3.8. *For any $j \in N \setminus I$, the inequality*

$$z_j + \sum_{i \in I_j} z_i \leq |I_j| \tag{66}$$

defines a facet of the convex hull of $\mathcal{K}$.

Proof. Proposition 3.7 gives an exact description of all the minimal covers of $\mathcal{K}$. The validity of (66) then follows immediately by observing that they are cover inequalities. Choose a

$j \in N \setminus I$. To show that the cover inequality (66) is facet-defining for j, we construct k affinely independent points that satisfy (66) at equality.

(p1) Choose a $l \in j \cup I_j$, set $z_l = 0$ and $z_i = 1$, for all $i \in j \cup I_j \setminus l$. Set $z_q = 0$, for all $q \in N \setminus (j \cup I_j)$. This gives us $|I_j| + 1$ points over all choices of l.

(p2) Let $z_k = 0$ and $z_i = 1$ for all $i \in j \cup I_j \setminus k$. Set $z_q = 1$ for some $q \in N \setminus (j \cup I_j)$ and all remaining variables to zero. This gives us $k - |I_j| - 1$ points over all choices of q.

By (64) the above points belong to $\mathcal{K}$. Also, they satisfy (66) at equality. It is straightforward to verify that these points are affinely independent. To show that the points are affinely independent, examine the matrix Ω whose columns are the points constructed in (p1) and (p2). We need to show that there does not exist a nonzero solution to the system of equations: $\Omega\alpha = 0, \sum_{i=1}^{k} \alpha_i = 0$. For the sake of contradiction, let α be a nonzero solution. For any $q \in N \setminus (j \cup I_j)$, there exists exactly one 1 in the q^{th} row of Ω and it corresponds to the column due to some point constructed in (p2). Hence, $\alpha_q = 0, \forall q \in N \setminus (j \cup I_j)$. Thus, we may focus only on the columns defined by points constructed in (p1). Observe that the submatrix formed by the columns defined by points constructed in (p1) is equal to $\begin{bmatrix} \mathbb{E} - \mathbb{I} \\ \mathbf{0} \end{bmatrix}$, where $\mathbb{E}$ is a matrix of ones, $\mathbb{I}$ is an identity and $\mathbf{0}$ is a matrix of zeros. Since these columns are linearly independent, we obtain $\alpha_q = 0, \forall q \in j \cup I_j$. Thus α is the zero vector, a contradiction. This completes the proof. □

Let us recall a previous result on $\{0,1\}$ knapsacks.

Lemma 3.1 (Balas and Jeroslow [17])**.** *The set of* $\{0,1\}$ *solutions to any knapsack inequality is equal to the set of* $\{0,1\}$ *solutions defined by extensions of its minimal covers, i.e.*

$$\tilde{\mathcal{K}} = \left\{ \tilde{z} \in \{0,1\}^n \colon \sum_{i \in E(\tilde{C})} \tilde{z}_i \leq |\tilde{C}| - 1, \text{ for all minimal covers } \tilde{C} \right\}.$$

Proposition 3.9. *The facet-defining cover inequalities* (66) *are sufficient to describe* $\mathrm{conv}(\mathcal{K})$*, i.e.,*

$$\mathrm{conv}(\mathcal{K}) = \left\{ z \in [0,1]^k \colon z_j + \sum_{i \in I_j} z_i \leq |I_j|, \; j \in N \setminus I \right\}. \tag{67}$$

Proof. The forward inclusion holds due to the fact that the cover inequalities are valid for $\text{conv}(\mathcal{K})$. Now assume w.l.o.g. that the cover inequalities are sorted from the smallest to the largest index $j \in N \setminus I$. Consider the matrix of coefficients defined by these cover inequalities. Note that for any $j \in N \setminus I$, the j^{th} column of the matrix has an entry 1 in exactly one row and all other entries are zero. Now consider the i^{th} column of the matrix for some $i \in I$. Let $J_i := \{j \in N \setminus I \colon j < i\}$. Then, using the characterization of minimal covers from Proposition 3.7, it follows that the i^{th} column has an entry 1 in exactly those rows that correspond to $j \in J_i$ and all other entries are zero. Since the rows were sorted to begin with, the nonzero entries in any column must occur successively. Thus, our coefficient matrix is an interval matrix (cf. [79]) and hence, totally unimodular (TU). Since adding bound constraints conserves the TU property of a matrix and right hand sides are integers, we obtain that the set defined by the inequalities (66) and bound constraints (called the minimal covering polytope) is integral.

We know from Proposition 3.7 that $k \in C$, for any minimal cover $C \in \mathcal{C}$. It follows that the extension of C to N is equal to $E(C) = C$. Using Lemma 3.1 gives us $\mathcal{K} = \{z \in \{0,1\}^k \colon \sum_{i \in E(C)} z_i \leq |C| - 1,\ \forall C \in \mathcal{C}\}$. Since the minimal covering polytope is integral and $E(C) = C$, the proof is complete. □

From Propositions 3.5, 3.6, and 3.7, and equation (58), we obtain the convex hull of $\mathcal{B}$.

Corollary 3.3.

$$\begin{aligned}\text{conv}(\mathcal{B}) = \Big\{(x,y,w,z,v) \colon\ & y = \sum_{i=1}^{k} 2^{i-1} z_i,\ w = \sum_{i=1}^{k} 2^{i-1} v_i, \\ & v_j + \sum_{i \in I_j} v_i - |I_j| x \leq 0, \quad j \in N \setminus I \\ & z_j - \frac{1}{a} v_j + \sum_{i \in I_j} \Big(z_i - \frac{1}{a} v_i\Big) + \frac{|I_j|}{a} x \leq |I_j|, \quad j \in N \setminus I \\ & v_i \geq 0, v_i \leq a z_i, v_i \leq x, v_i \geq x + a z_i - a, i \in \{1, \ldots, k\}\Big\}.\end{aligned} \tag{68}$$

3.3.3 Some extensions

We next address some closely related cases.

Multiple bilinear terms. Consider a set with multiple bilinear terms corresponding to a single continuous variable x. In particular, we consider bilinear terms of the form $w_j = xy_j$ for $j = 1, \ldots, n$. The reformulated set in (x, z, v)-space is

$$\mathcal{S} = \Big\{(x, z, v) \in \quad \Re_+ \times \Re^{\sum_{j=1}^n k_j} \times \Re^{\sum_{j=1}^n k_j} \colon z^j \in \mathcal{K}_j,\ j = 1, \ldots, n \\ v_i^j \geq 0, v_i^j \leq a z_i^j, v_i^j \leq x, v_i^j \geq x + a z_i^j - a, \forall i \in \{1, \ldots, k\}, j \in \{1, \ldots, n\}\Big\}.$$

Proposition 3.4 implies that the convex hull of $\mathcal{S}$ is given by the inequalities from Proposition 3.5 corresponding to each z^j and v^j, for $j = 1, \ldots, n$.

Semiinteger variables. Let y be a semiinteger, i.e. $y \in \{0\} \cup \{b', b' + 1, \ldots, b\}$ for some positive integers b', b. Rewriting $y = b' z_0 + \sum_{i=1}^k 2^{i-1} z_i$ yields

$$\mathcal{S}' = \left\{ z \in \{0,1\}^{k+1} \colon \sum_{i=1}^k 2^{i-1} z_i \leq b - b',\ z_i \leq z_0, \forall i \in \{1, \ldots, k\} \right\}, \tag{69}$$

where $y \in \{b', b' + 1, \ldots, b\}$ if and only if $z_0 = 1$. Represent $\mathcal{K}'$ and $\text{conv}(\mathcal{K}')$ as

$$\begin{aligned} \mathcal{K}' &= \{z \in \{0,1\}^k \colon \sum_{i=1}^k 2^{i-1} z_i \leq b - b'\} \\ \text{conv}(\mathcal{K}') &= \{z \in \Re^k \colon \Pi z \leq \pi_0\} \end{aligned}$$

$\text{conv}(\mathcal{K}')$ can be obtained from Proposition 3.9.

Observe that $\mathcal{S}' = (\mathcal{K}' \times \{1\}) \cup \{\mathbf{0}\}$. Hence,

$$\begin{aligned} \text{conv}(\mathcal{S}') &= \text{conv}(\text{conv}(\mathcal{K}' \times \{1\}) \cup \{\mathbf{0}\}) \\ &= \text{conv}(\{z \in \Re^{k+1} \colon \Pi z \leq \pi_0, z_0 = 1\} \cup \{\mathbf{0}\}). \end{aligned}$$

Disjunctive programming provides an extended formulation that can be easily projected to obtain the identity

$$\text{conv}(\mathcal{S}') = \{z \in \Re^{k+1} \colon \Pi z \leq \pi_0 z_0\}.$$

Applying Proposition 3.5 gives the convex hull of the corresponding mixed semiinteger bilinear set.

Missing powers of 2. Suppose that in the binary expansion knapsack defined in equation (56), some powers of two are missing. Thus we have

$$\mathcal{K}' := \left\{ z \in \{0,1\}^{k'} : \sum_{t=1}^{k'} 2^{i_t - 1} z_t \leq b \right\}, \tag{70}$$

where $\{i_1, i_2, \ldots, i_{k'}\} \subseteq \{1, 2, \ldots, k\}$ such that $i_{k'} = k$. Note that if $i_{k'} < k$, then the knapsack inequality is redundant. The minimal covering inequalities for $\mathcal{K}'$ can be obtained using Proposition 3.7 with the added restriction that $z_i = 0$, for $i \notin \{i_1, i_2, \ldots, i_{k'}\}$. The knapsack weights still form a superincreasing sequence and hence the minimal covers of $\mathcal{K}'$ define its convex hull. (Alternatively, the corresponding coefficient matrix is still TU and defines $\text{conv}(\mathcal{K}')$). This observation has the following implication for a branch-and-cut algorithm that branches on the z variables.

Observation 3.2. *After adding covering inequalities from Proposition 3.9 as cutting planes at the root node, we cannot obtain any nontrivial cover cuts corresponding to $\mathcal{K}$ at nodes below the root node.*

Product of two integer variables. Suppose that in the definition of $\mathcal{X}$ in (52) we further restrict $x \in \mathbb{Z}_+$. If we perform binary expansion on only y, then the result of Proposition 3.5 carries through by observing that $x \in \{0, a\}$ at extreme points. Hence, we can use minimal covering inequalities and multiply them with x and $a - x$ to obtain $\text{conv}(\mathcal{B})$ as in Corollary 3.3. Now, if we perform binary expansion of both x and y, then Corollary 3.3 provides valid inequalities but not necessarily the convex hull of $\mathcal{B}$. This set was studied in Günlük et al. [54] after relaxing the knapsacks corresponding to x and y, and it was shown that McCormick linearizations are sufficient to characterize the convex hull of this relaxation.

General expansion of y. The binary expansion knapsack $\mathcal{K}$ can be viewed as a special case of the α-nary expansion general integer knapsack

$$\mathcal{K}(\alpha, b) := \left\{ z \in \mathbb{Z}_+^n : \sum_{i=1}^{n} \alpha^{i-1} z_i \leq b,\ 0 \leq z_i \leq \alpha - 1, \forall i \right\},$$

with $\alpha = 2$. We discuss the convex hull of $\mathcal{K}(\alpha, b)$ in §3.6.

3.4 Computational results

3.4.1 Experimental setup

In this section, we report computational results on several test instances. Given a mixed integer bilinear problem, we solved it using the open source nonconvex MINLP solver `Couenne` 0.3 [21]. Our goal is to test whether these bilinear problems can be solved efficiently using the MILP formulations (M-MIBLP) and (B-MIBLP) from §3.2. We used `CPLEX 12.1` as the MILP solver. Since `Cplex` is a sophisticated commercial MILP solver whereas `Couenne` is a relatively new open source MINLP solver, we cannot and do not wish to draw conclusions regarding the performance of the spatial branch-and-bound algorithm. Instead, our aim is to show that the proposed MILP approach is a viable alternative on certain classes of problems.

To ensure numerical consistency between `Couenne` and `Cplex`, we used the following algorithmic parameters: `feasibility tolerance` = 10^{-6}, `integrality tolerance` = 10^{-5}, `relative optimality gap` = 0.01%, `absolute optimality gap` = 10^{-4}. Additionally, for `Cplex`, we set `Threads = 1` to enforce single threaded computing. All other options were set to default values for the respective solver. Our assumption of nonnegative lower bounds on variables is without any loss of generality since we translated every variable with a nonzero lower bound so that the formulation conforms to (MIBLP1).

For the MILP relaxation (M-MIBLP), we employed branching on integer solutions, as discussed next.

Branching strategy for solving (M-MIBLP)**.** One way of obtaining the true optimum value η^*(MIBLP) using formulation (M-MIBLP) is to branch on integer feasible solutions. Suppose that we are at a node in the MILP search tree such that the solution at this node $(x_l^*, y_j^*, w_{lj}^*) \in \mathcal{M}_{lj} \setminus \mathcal{X}_{lj}$, for some indices l, j. Then, we can branch on the variable y_j using the disjunction $y_j \leq y_j^* \vee y_j \geq y_j^* + 1$. Note that if y_j is marked as a candidate for branching, then it must be that $y_j^* \in (0, u_j)$ since the McCormick linearization $\mathcal{M}_{lj}$ of $\mathcal{X}_{lj}$ is exact when $y_j \in \{0, u_j\}$. After branching on y_j, the McCormick envelopes of $w_{lj} = x_l y_j$ in the two branches are updated using the refined bounds on y_j. An integer feasible node is

accepted as an incumbent solution when $|w^*_{lj} - x^*_l y^*_j| \leq \epsilon, \forall\, l \in \{1, ..., m\},\ j \in \{1, ..., n\}$, for a small enough positive ϵ. Hence, at termination, we obtain an optimal solution to (MIBLP). To ensure numerical correctness of the algorithm, the value of ϵ should be chosen equal to the feasibility tolerance in the MILP solver. It is important to observe here that in this proposed branching strategy, we only branch on some integer variable y_j. Thus while solving (M-MIBLP), we do not branch on a continuous variables x_l as done in the spatial branch-and-bound framework within global optimization solvers.

While branching at any fractional or integer node of the branch-and-bound tree for (M-MIBLP), updated McCormick envelopes were added for each bilinear term corresponding to the branching variable, using the local bounds on the variables at this particular node. This is a standard technique used by global optimization solvers. In our preliminary computations, this technique performed better than updating the envelopes only when we branch on integer nodes. The variable selection rule for branching on integer nodes was based on maximum violated bilinear term whereas for fractional nodes we used the branches proposed by `Cplex`. By solving the relaxation (M-MIBLP) as a MILP without branching on continuous variables, we have adopted a traditional branch-and-bound solution strategy to test if branching on integer nodes in original space can outperform the spatial branch-and-bound algorithm of `Couenne` or the extended binary MILP reformulation (B-MIBLP).

While solving reformulation (B-MIBLP), the general integer variables $y_j, \forall j \in \{1, ..., n\}$, were substituted out in order to reduce the problem size and to ensure that branching is performed solely on the binary variables. One approach was to solve this reformulation using default branch-and-cut options for `Cplex`. In the second approach, we added all the inequalities defining $\text{conv}(\mathcal{B}_{lj})$, for all $l \in \{1, ..., m\},\ j \in \{1, ..., n\}$ (see (68)), to the user cut pool of `Cplex` along with default branch-and-cut options. Observation 3.2 tells us that these covering inequalities along with the branched upon variables imply convex hull of (56) at nodes below the root node. In our preliminary computations, we also tested the following idea: retaining integer variables $y_j, \forall j$, and whenever `Cplex` chooses some y_j as a branching variable, adding cover inequalities corresponding to the refined bound on y_j as local cuts at this node. However, we found no performance gain with this approach.

The experiments were carried out on three types of instances and run on a Linux machine with kernel 2.6.18 running on a 64-bit x86 processor and 32GB of RAM. The time limit was 1 hour barring a few instances. Tables 2–13 highlight comparisons between four solution approaches - `Couenne`, (M-MIBLP), (B-MIBLP) + Cuts, and (B-MIBLP). We report the number of nodes (Nds) processed by the branch-and-bound algorithm and the running time (T) in seconds. A * indicates the instance was not solved to optimality within the time limit. For the binary expansion reformulation, we also report the total number of cover inequalities that were separated by `Cplex` (Cuts) and the % root gap closed (Rgp-cl) by adding our cuts with `Cplex` cuts over adding only `Cplex` cuts.

For an instance $\mathcal{I}$ not solved to optimality, we report two types of % optimality gaps. The first one is the % optimality gap of a algorithm $\mathcal{A}$, given by

$$\mu_{\mathcal{I}}(\mathcal{A}) = 100 \times \left| 1 - \frac{\text{Best LB by } \mathcal{A}}{\text{Best Feas by } \mathcal{A}} \right| \tag{71}$$

Thus, $\mu_{\mathcal{I}}(\mathcal{A})$ denotes how close the algorithm $\mathcal{A}$ was to solving $\mathcal{I}$ to optimality. The second metric is the % optimality gap in terms of the best feasible solution found for $\mathcal{I}$ across all algorithms, given by

$$\omega_{\mathcal{I}}(\mathcal{A}) = 100 \times \left| 1 - \frac{\text{Best Feas by } \mathcal{A}}{\text{Best Feas for } \mathcal{I}} \right| \tag{72}$$

An optimality gap of (–) means no integer feasible solution was found by the algorithm within the time limit.

3.4.2 General mixed integer bilinear problems

This set of instances contains problems formulated as (MIBLP1) where bilinear terms are present in both the objective function and constraints. We divide into two subcategories depending on the source of the test problems.

3.4.2.1 `MINLPLib`

We chose 14 instances from this test library [28] that have bilinearities between continuous and integer variables, such as xy, or between two integer variables, such as y_1y_2. Note that for a bilinear term y_1y_2 where $y_i \in \mathbb{Z}_+, i = 1, 2$, the result of Proposition 3.5 carries through. Instances `lop97ic` and `lop97icx` are not considered because of their large size. Instances

`tln2` - `tloss` are the bilinear version of the cutting stock problem, where the number of rolls produced by each cutting pattern is also an integer variable.

Table 2: Test instances from `MINLPLib`

Instance	`Couenne`		M-MIBLP		B-MIBLP + Cuts				B-MIBLP	
	Nds	T	Nds	T	Nds	T	Cuts	Rgp-cl	Nds	T
`ex1263a`	1121	2	2366	1	635	1	0	0	640	1
`ex1264a`	940	1	2762	1	519	1	0	0	519	1
`ex1265a`	197	3	995	1	378	1	0	0	378	1
`ex1266a`	61	1	562	1	10	1	0	0	10	1
`prob02`	0	0	170	1	12	1	0	0	12	0
`prob03`	0	0	4	1	0	0	1	5	0	0
`tln2`	2	0	12	1	183	0	0	0	183	0
`tln4`	47384	55	98770	118	4401	4	17	0	4576	4
`tln5`	496377	*	306394	*	12662	17	0	0	12662	18
`tln6`	421402	*	242486	*	56130	87	102	0	65514	93
`tln7`	316152	*	684937	*	1249707	*	36	0	1728487	*
`tln12`	96500	*	50618	*	134823	*	132	0	180712	*
`tloss`	537	3	877	1	84	1	0	0	84	1
`tltr`	371	1	144	1	214	1	11	2	181	1

Table 3: Optimality gaps for test instances from `MINLPLib`

Instance	`Couenne`		M-MIBLP		B-MIBLP + Cuts		B-MIBLP	
	$\mu_{\mathcal{I}}$	$\omega_{\mathcal{I}}$	$\mu_{\mathcal{I}}$	$\omega_{\mathcal{I}}$	$\mu_{\mathcal{I}}$	$\omega_{\mathcal{I}}$	$\mu_{\mathcal{I}}$	$\omega_{\mathcal{I}}$
`tln5`	43	2	44	3	0	0	0	0
`tln6`	54	0	69	1	0	0	0	0
`tln7`	78	17	77	6	34	0	1	0
`tln12`	–	–	–	–	81	0	81	2

From Table 2 we observe that the bilinear cutting stock instances `tln4` - `tln12` are perhaps the most difficult ones from this set of instances. On these 5 instances, the binary reformulation, with or without our cuts, has done better than both envelope relaxation (M-MIBLP) and solving with `Couenne`. In particular, for `tln4`, the nodes and time taken by binary MILP was substantially less than for the other two, whereas `tln5` and `tln6` were solved within the time limit by binary MILP (with some help from cuts on `tln6`). Although, `tln7` and `tln12` remained unsolved by all four methods, the optimality gap at termination

was higher for the first two methods.

3.4.2.2 Product bundling

The product bundling problem, addressed in [44], can be defined as follows: let P be a set of products and C be a set of customers. The variable $x_p \in \mathbb{Z}_+$ represents the number of units of product p in a bundle and $y_c \in \mathbb{Z}_+$ represents the number of bundles bought by customer c. The objective is to maximize $\sum_{c\in C}\sum_{p\in P} x_p y_c$, which is the total number of products bought, subject to the demand constraint $x_p y_c \leq D_{cp}, \forall c \in C, p \in P$. Here, $D_{cp} \in \mathbb{Z}_+$ and not all D_{cp} are zero. Thus, the formulation is

$$\max\left\{\sum_{c\in C}\sum_{p\in P} x_p y_c \colon x_p y_c \leq D_{cp},\ x_p, y_c \in \mathbb{Z}_+ \forall c, p\right\}. \tag{Bundling}$$

Clearly, $x \geq \mathbf{0}$ and $y \geq \mathbf{0}$. We first obtain valid upper bounds on the variables.

Proposition 3.10. *The variables x and y in* (Bundling) *can be upper bounded as*

$$\begin{aligned} x_p &\leq \max\{D_{cp} \colon c \in C\}, \qquad p \in P \\ y_c &\leq \max\{D_{cp} \colon p \in P\}, \qquad c \in C \end{aligned}$$

Proof. We first claim that the optimal value of (Bundling) is at least 1. Since not all D_{cp} are zero and $D_{cp} \in \mathbb{Z}_+, \forall c, p$, there must be exist some $c \in C, p \in P$ such that $D_{cp} \geq 1$. Set $x_p = y_c = 1$ and all other variables zero. This is a feasible solution with objective value 1.

We now show that any optimal solution (x^*, y^*) to (Bundling) must satisfy $x^* \leq \hat{x}$ and $y^* \leq \hat{y}$, where $\hat{x}$ and $\hat{y}$ are the proposed upper bounds. Suppose that $x^*_p > \hat{x}_p$ for some $p \in P$. This implies that $y^*_c = 0$, for all $c \in C$, since every feasible point must satisfy $x_p y_c \leq D_{cp}, \forall c$. Hence, the optimal value must be zero, which is a contradiction to our first claim. Similarly for y^*. Hence, $\hat{x}$ and $\hat{y}$ are valid upper bounds that do not cut off any points from the set of optimal solutions. □

Our first problem set of this type consists of 54 instances, created using the random generator of [44]. Half of these are for $|C| = 10, |P| = 30$ and the other half for $|C| = 20, |P| = 50$. For each problem size, we considered $\rho \in \{0.2, 0.5, 0.8\}$ and $\lambda \in \{30, 100, 200\}$, where $D_{cp} = 0$ with probability ρ and if $D_{cp} > 0$, then $D_{cp} \sim$ Poisson(λ). For each

combination of ρ and λ, 3 instances were created. Note that a bilinear term $w_{cp} = x_p y_c$ exists only if $D_{cp} > 0$. Otherwise $x_p = 0 \vee y_c = 0$. This disjunction is modeled as a bigM constraint using extra binary variables for (M-MIBLP) whereas for (B-MIBLP), the condition $w_{cp} = 0$ is incorporated in the McCormick linearization. As λ increases, the set of integer feasible solutions increases and as ρ decreases, the demand matrix becomes more dense giving rise to more bilinear terms.

Table 4: Product bundling instances : test set 1. $|C| = 10, |P| = 30$. 27 random instances.

	`Couenne`	M-MIBLP	B-MIBLP + Cuts	B-MIBLP
Average Nds	388824	735621	349335	383832
Average T (sec.)	2103.17	2409.01	2787.38	2735.84
Average Cuts	–	–	517	–
Average % Rgp-cl	–	–	0.5	–
(a) # solved	13	9	9	8
(b) # fastest optimal	10	4	0	0
(c) Average time gain (sec.)	880.70	18.25	0.00	0.00
(d) # best feasible	1	10	2	4
(e) Average $\mu_{\mathcal{I}}$	98.94	20.43	211.51	212.94
(f) Average $\omega_{\mathcal{I}}$	13.05	4.22	1.44	1.78

In Tables 4 and 5, we present average values over the 27 random instances for each problem size. We chose a time limit of 1 hour, since a longer time limit allows a better explanation of the average values over all the instances. We report the average values for our metrics - number of nodes, time taken (sec.), number of user cuts added, and % root gap closed, where the averages are taken over instances in each subgroup. For each method, we also provide the a) number of instances it solved to optimality (# solved), b) number of instances it found an optimality certificate in the shortest amount of time (# fastest optimal), and c) average time it was faster than the next best algorithm on the instances in (b) (Average time gain). Since there exist some instances that are not solved to optimality by any of the formulations, in addition to $(a)-(c)$, we also report the following metrics over the instances that remained unsolved with any of the formulations: d) number of instances on which the best feasible solution was found (# best feasible), e) average $\mu_{\mathcal{I}}$, and f) average

$\omega_{\mathcal{I}}$.

From both the tables we observe that `Couenne` solved the most number of instances in 1hr. However, amongst the unsolved problems, the best feasible solutions obtained from binary reformulation helped produce strong lower bounds (since its maximization) on the problem. This can be concluded by comparing the optimality gaps $\omega_{\mathcal{I}}$ for the four different methods. In Table 4, (M-MIBLP) was able to produce the best feasible solution on the most number of instances (10). However, the relative quality of these solutions, denoted by $\omega_{\mathcal{I}}$, was weaker than (B-MIBLP) (with and without cuts) implying that the solutions obtained with the binary reformulation model were either the best or very close to being the best. For the larger problem sizes in Table 5, a similar reasoning holds for the $\omega_{\mathcal{I}}$ values along with the fact that now the best feasible solutions were obtained solely by one of the two binary models. On the relaxation side, it seems that although a large number of our cover cuts were separated, they were not effective in closing the root gap. In fact, most of our user cuts were separated deeper in the branch-and-cut tree suggesting that the default cuts added by `Cplex` at root node were itself quite strong on these instances. (M-MIBLP) has the lowest average termination gap $\mu_{\mathcal{I}}$ and for this set of instances, our proposed branching strategy performed fairly well, possibly due to not too large interval width of the general integers.

Table 5: Product bundling instances : test set 2. $|C| = 20, |P| = 50$. 27 random instances.

	`Couenne`	M-MIBLP	B-MIBLP + Cuts	B-MIBLP
Average Nds	99790	241450	221272	181680
Average T (sec.)	2776.51	3600.00	3600.00	3600.00
Average Cuts	–	–	1245	–
Average % Rgp-cl	–	–	0.3	–
(a) # solved	8	0	0	0
(b) # fastest optimal	8	0	0	0
(c) Average time gain (sec.)	2788.84	0.00	0.00	0.00
(d) # best feasible	0	0	10	9
(e) Average $\mu_{\mathcal{I}}$	2290.93	343.71	623.54	622.45
(f) Average $\omega_{\mathcal{I}}$	51.04	21.10	3.88	4.48

Table 6: The `watts` instances for product bundling.

Instance	Couenne		M-MIBLP		B-MIBLP + Cuts				B-MIBLP	
	Nds	T	Nds	T	Nds	T	Cuts	Rgp-cl	Nds	T
5x41	305800	*	1217175	*	25893	141	18	0	33762	176
5x41m	1044023	*	1301411	*	23323	166	23	0	16426	226
9x60	120260	*	319347	*	55277	*	745	0	86562	*
10x60	97180	*	239071	*	74913	*	805	0	69911	*
10x60d	112030	*	291302	*	31753	808	234	0	60945	1902

The second set of product bundling problems consists of 5 instances from a real food company, as used in [44]. These are referred to as the `watts` instances, reported in Tables 6 and 7. For these five instances, we clearly see that the binary reformulation is superior, both in terms of $\mu_{\mathcal{I}}$ and $\omega_{\mathcal{I}}$ and the solved instances. Although our cuts were not effective at the root node, they were helpful on expediting the solve of 3 out of the 5 instances, especially `10x60d` whose solution time was more than halved. On the contrary, for `9x60` and `10x60`, a lot of user cuts were separated below the root node, which potentially led to slow down of `Cplex` and hence a higher termination gap than (B-MIBLP) without cuts.

Table 7: Optimality gaps for `watts` instances

Instance	Couenne		M-MIBLP		B-MIBLP + Cuts		B-MIBLP	
	$\mu_{\mathcal{I}}$	$\omega_{\mathcal{I}}$	$\mu_{\mathcal{I}}$	$\omega_{\mathcal{I}}$	$\mu_{\mathcal{I}}$	$\omega_{\mathcal{I}}$	$\mu_{\mathcal{I}}$	$\omega_{\mathcal{I}}$
5x41	40	7	165	24	0	0	0	0
5x41m	6	0	295	25	0	0	0	0
9x60	461	22	277	58	111	0	65	0
10x60	338	24	252	60	77	0	59	0
10x60d	192	17	157	46	0	0	0	0

3.4.3 Nonconvex objective function with linear constraints

3.4.3.1 MIPLIB

We chose MILP instances from `MIPLIB` 2003 and modified the objective function to a bilinear function. Thus, the feasible region for these instances is a polyhedron and all nonconvexities appear in the objective. For general MILPs, only those instances with less than 1000 integer

and 1000 continuous variables were selected and the objective was

$$\max \sum_i y_i(x_i + x_{i+1} + x_{i+2}). \tag{73}$$

Here x_i and y_i are bounded continuous and integer variables, respectively, and the indexing of these variables is as determined by `Cplex` after importing the `.mps` input file for the `MIPLIB` instance. The summation in (73) is taken only over those variables which were either originally bounded or their LP based bounds (maximizing and minimizing each variable over the LP relaxation of the feasible set) were finite.

Table 8: General MILP test instances from `MIPLIB`

Instance	`Couenne`		M-MIBLP		B-MIBLP + Cuts				B-MIBLP	
	Nds	T	Nds	T	Nds	T	Cuts	Rgp-cl	Nds	T
`arki001`	1497	*	47635	*	50233	*	131	7	20457	*
`noswot`	98018	*	213037	*	4398	5	0	0	4398	5
`gesa2`	46	124	0	1	0	1	0	0	0	1
`gesa2-o`	261	*	54322	*	69	3	782	99	36644	*
`rout`	87613	*	44	1	50	1	5	0	31	1
`timtab1`	37294	*	291471	*	311058	*	63	7	339376	*
`timtab2`	48624	*	136749	*	133526	*	136	6	138606	*
`roll3000`	3	*	42147	*	28678	461	28	1	27649	496

Table 9: Optimality gaps for test instances from `MIPLIB`

Instance	`Couenne`		M-MIBLP		B-MIBLP + Cuts		B-MIBLP	
	$\mu_{\mathcal{I}}$	$\omega_{\mathcal{I}}$	$\mu_{\mathcal{I}}$	$\omega_{\mathcal{I}}$	$\mu_{\mathcal{I}}$	$\omega_{\mathcal{I}}$	$\mu_{\mathcal{I}}$	$\omega_{\mathcal{I}}$
`arki001`	–	–	21	8	5	0	7	1
`noswot`	12	20	46	27	0	0	0	0
`gesa2-o`	–	–	1	0	0	0	3	0
`rout`	5	5	0	0	0	0	0	0
`timtab1`	38	10	24	7	9	0	10	0.2
`timtab2`	–	–	47	11	28	0	29	0.05
`rol3000`	–	–	92	63	0	0	0	0

Only three out of the total eight instances remained unsolved for B-BLP + Cuts, least amongst all four methods. For `arki001`, `timtab1`, and `timtab2`, our cuts seemed helpful in closing some gap at the root node. For `gesa2-o`, our cuts helped solve the problem very

quickly. Observe that for `arki001`, `gesa2-o`, `timtab2`, `roll3000`, `Couenne` was unable to find a integer feasible solution within the time limit and in fact, could process only three nodes for `roll3000`, likely because of the large number of general integers in this instance. On these same four instances, our cuts were either able to solve the binary reformulation or could reduce the optimality gap.

3.4.3.2 BoxQP

Here we consider box constrained nonconvex quadratic problems

$$\begin{aligned} \min \quad & \tfrac{1}{2}x^\top Qx + f_0^\top x \\ \text{s.t.} \quad & x \in [0, \tilde{u}] \cap \mathbb{Z}^n_+. \end{aligned} \tag{Integer BoxQP}$$

Introducing a new continuous variable $y = Qx$, we can rewrite the above problem with a bilinear objective and linear constraints as

$$\begin{aligned} \min \quad & \tfrac{1}{2}x^\top y + f_0^\top x \\ \text{s.t.} \quad & y = Qx \\ & y_i^L \le y_i \le y_i^U, \quad i = 1, \dots, n \\ & x \in [0, \tilde{u}] \cap \mathbb{Z}^n_+. \end{aligned} \tag{Bilinear Integer BoxQP}$$

where $y_i^L := \sum_{j\colon q_{ij}<0} q_{ij}\tilde{u}_j$ and $y_i^U := \sum_{j\colon q_{ij}>0} q_{ij}\tilde{u}_j$, for $i = 1, \dots, n$. In this transformed problem, every bilinear term $w_i = x_i y_i, i = 1, \dots, n$, is a product between a bounded integer variable and a bounded continuous variable and hence conforms to the assumptions of this study.

The test instances for our computational experiments were obtained from the 54 random instances of Vandenbussche and Nemhauser [106], where the authors studied $[0, 1]$ constrained nonconvex QPs. The value of n, i.e. the number of variables in (Integer BoxQP), lies in $\{20, 30, 40, 50, 60\}$ for these instances. For every instance of $[0, 1]$ box QP, we generated values of integral upper bounds $\tilde{u}_i$ uniformly at random between 10 and 100, for all i. Then after a suitable scaling of the coefficient matrix and cost vector with these upper bounds, we obtain an instance for (Integer BoxQP).

The results of our experiment are summarized in Table 10 and 11. The second column in Table 10 corresponds to the solution of the the reformulated bilinear problem

(Bilinear Integer BoxQP) with `Couenne`. Of the unsolved instances, the average values of $\omega_{\mathcal{I}}$ are lowest for the two binary formulations, indicating that good quality solutions are obtained by solving the MILP formulation. Our cuts close around 41% of the root gap, which translates into lower termination gap $\mu_{\mathcal{I}}$, for e.g. $42.94 < 66.38$ in Table 10, and helps `Cplex` spend more time in obtaining good feasible solutions for the most number of unsolved instances, 41 out of 52 in Table 10.

Table 10: 54 instances of (Integer BoxQP) from Vandenbussche and Nemhauser [106] where `Couenne` is solved as (Bilinear Integer BoxQP).

	`Couenne` (Bilinear Integer BoxQP)	M-MIBLP	B-MIBLP + Cuts	B-MIBLP
Average Nds	850670	222470	433045	1056528
Average T (sec.)	3501	3600	3491.17	3587.80
Average Cuts	–	–	113	–
Average % Rgp-cl	–	–	41	–
(a) # solved	2	0	2	1
(b) # fastest optimal	2	0	0	0
(c) Average time gain (sec.)	594	0	0	0
(d) # best feasible	7	1	41	34
(e) Average $\mu_{\mathcal{I}}$	33.74	117.85	42.94	66.38
(f) Average $\omega_{\mathcal{I}}$	1.54	8.62	0.21	0.15

We compare the performance of `Couenne` on (Integer BoxQP) and (Bilinear Integer BoxQP) in Table 11. Notice that there is a significant degradation in the performance of `Couenne` when the model is (Bilinear Integer BoxQP). The number of solved instances (a) and number of unsolved instances on which `Couenne` found the best feasible solution (d) reduces drastically between the two columns in Table 11. This further indicates that the presence of bilinear terms in the objective function (only) can be a great source of difficulty for an MINLP solver.

3.4.3.3 Disjoint bilinear problems

Using the instance generator of Vicente et al. [107], 100 random instances of disjoint bilinear problems were created. These test instances have a bilinear objective function and the

Table 11: Comparing performance of `Couenne` on (Integer BoxQP) and (Bilinear Integer BoxQP).

	`Couenne` (Integer BoxQP)	`Couenne` (Bilinear Integer BoxQP)
Average Nds	177312	850670
Average T (sec.)	2852.53	3501
(a) # solved	14	2
(b) # fastest optimal	14	2
(c) Average time gain (sec.)	2733.64	594
(d) # best feasible	12	7
(e) Average $\mu_{\mathcal{I}}$	45.08	33.74
(f) Average $\omega_{\mathcal{I}}$	2.09	1.54

feasible region is defined by a cartesian product of two polyhedra, one in x-space and another in y-space. The y variables are restricted to be integer.

$$\begin{aligned} \min \quad & x^\top Q_0 y + f_0^\top x + g_0^\top y \\ \text{s.t.} \quad & x \in X := \{x \in \Re^{2\kappa_2} \colon Ax \leq h_a\} \\ & y \in Y := \{y \in \mathbb{Z}^{\kappa_1+\kappa_2} \colon By \leq h_b\}. \end{aligned} \tag{Disjoint BLP}$$

The values $\delta = 2$ and $\rho = 0$ were used while generating components of matrices A and B and the final values of $Q_0, f_0, g_0, A, B, h_a, h_b$ were obtained using randomized Householder matrices, the seed for which was set equal to $instanceid \times \kappa_1 \times \kappa_2$. A more detailed description of the instance generator can be found in [107]. The parameters κ_1 and κ_2 control the size of the problem. The total number of variables and constraints in (Disjoint BLP) is equal to $\kappa_1 + 3\kappa_2$ and $2\kappa_1 + 4\kappa_2$, respectively. LP based bounds are generated for each variable and any unbounded variable is given a artificial upper (lower) bound of 100 (-100).

The instances are divided into two subgroups: half of them were generated with $\kappa_1 = 2, \kappa_2 = 4$ and the other half for $\kappa_1 = 3, \kappa_2 = 5$.

In Table 12, all the methods, except (M-MIBLP), were able to solve all 50 instances to optimality. The binary formulation with user cuts was fastest on 60% of the instances. This was primarily because the minimal cover inequalities closed about 34% of the root gap.

Table 12: Disjoint bilinear instances : $\kappa_1 = 2, \kappa_2 = 4$. 50 random instances.

	`Couenne`	M-MIBLP	B-MIBLP + Cuts	B-MIBLP
Average Nds	24289	78414	44322	44090
Average T (sec.)	45.22	3351.21	23.96	22.60
Average Cuts	–	–	96	–
Average % Rgp-cl	–	–	34.18	–
(a) # solved	50	6	50	50
(b) # fastest optimal	12	0	30	15
(c) Average time gain (sec.)	23.04	0.00	1.92	1.65

`Couenne` was fastest on only 24% of the instances. Note that there exist some instances on which more than one method solved in shortest time. The average time gained by faster performance of binary with cuts (1.92 sec.) was not as high as that gained in `Couenne` (23.04 sec.). This is possibly due to the two binary methods: with and without user cuts, performing almost equally well in `Cplex`. Hence, for each instance that was solved quickest by either of the two binary methods, we also computed the time gained by the best binary method. The average value of this metric was found to be 41.46 sec., higher than the average time gain of 23.04 sec. in `Couenne`.

Table 13: Disjoint bilinear instances : $\kappa_1 = 3, \kappa_2 = 5$. 50 random instances.

	`Couenne`	M-MIBLP	B-MIBLP + Cuts	B-MIBLP
Average Nds	114430	5110	168612	185594
Average T (sec.)	3424.85	3605.00	2565.84	2773.91
Average Cuts	–	–	172	–
Average % Rgp-cl	–	–	33.31	–
(a) # solved	16	0	21	8
(b) # fastest optimal	8	0	14	3
(c) Average time gain (sec.)	688.58	0.00	1154.73	936.12
(d) # best feasible	6	0	19	16
(e) Average $\mu_{\mathcal{I}}$	2.15	21.55	2.21	2.92
(f) Average $\omega_{\mathcal{I}}$	0.035	0.285	0.003	0.006

The instances in Table 13 are larger in size due to the higher values of κ_1 and κ_2.

Once again, the binary formulation with user cuts solved the most number of instances to optimality and also in shortest time (28%). In this case, the average time gained by this faster performance was significantly higher than the other three (1155 sec). Of the 25 instances that remained unsolved after 1 hour by any of the formulations, although the binary formulation (both with and without cuts) produced the best feasible solution on more than half of them, the values of the best feasible solutions produced by `Couenne` were almost as good. This is indicated by the low average values of $\omega_{\mathcal{I}}$ for these three formulations.

3.5 Bounded bilinear terms

In this section, we consider the single mixed integer bilinear set $\mathcal{X}$ from equation (52) along with a nontrivial upper bound on the product xy. Define this set to be

$$\mathcal{X}_u := \{(x, y, w) \in \Re_+ \times \mathbb{Z}_+ \times \Re_+ \colon w = xy, xy \leq u, x \leq a, y \leq b\}. \tag{74}$$

Here $0 < u < ab$. Note that the projection of $\mathcal{X}_u$ onto the (x, y)-space can be convexified using a single secant

$$(b - \lfloor u/a \rfloor)\left(x - \frac{u}{b}\right) + \left(a - \frac{u}{b}\right)(y - b) \leq 0.$$

that joins the two end points $(a, \lfloor u/a \rfloor)$ and $(u/b, b)$.

The binary reformulation of $\mathcal{X}_u$ (after eliminating y and w) is

$$\begin{aligned}\mathcal{B}_u := \Big\{(x, z, v) \in \Re_+ \times \{0,1\}^k \times \Re_+^k \colon & \sum_{i=1}^{k} 2^{i-1} z_i \leq b,\ x \leq a, \\ & x\left[\sum_{i=1}^{k} 2^{i-1} z_i\right] \leq u,\ v_i = x z_i, i = 1, \ldots, k\Big\}.\end{aligned} \tag{75}$$

We first show that the product term $x \sum_{i=1}^{k} 2^{i-1} z_i \leq u$ has a combinatorial interpretation, by proving the following.

Proposition 3.11. *Let $K := \{1, \ldots, k\}$ and for any subset $S \subseteq K$ and coefficients $\sigma \in \Re_{++}^k$, denote $\sigma(S) = \sum_{i \in S} \sigma_i$. Define a set function $f \colon 2^k \mapsto \Re_-$ as*

$$f(S) = \begin{cases} f_0 & S = \emptyset, \\ -u/\sigma(S) & S \neq \emptyset. \end{cases} \tag{76}$$

for some $f_0 \leq 0$. Then, there exists some finite $f_0 \leq 0$ for which f is a nondecreasing, submodular function.

Proof. By concavity of the negative reciprocal function over $\Re_{++}$ and since $u \geq 0$, it follows that the difference function $f(S \cup i) - f(S)$ is nonincreasing, and hence f is submodular over $2^k \setminus \emptyset$. To preserve submodularity over 2^k, Nemhauser and Wolsey [79], Proposition III.3.2.1 implies f_0 must be such that

$$\begin{aligned} f(i) - f_0 &\geq f(i \cup j) - f(j), \qquad \forall \{i,j\} \subset K, i \neq j \\ \iff \quad f_0 &= u \min_{i,j} \left\{ \frac{1}{\sigma_i + \sigma_j} - \frac{1}{\sigma_i} - \frac{1}{\sigma_j} \right\}. \end{aligned}$$

Since $h(\chi) = 1/(\lambda + \chi) - 1/\chi$ is monotone over $\Re_{++}$ for $\lambda > 0$, the minimum in f_0 is attained at $i = \sigma_{(1)}, j = \sigma_{(2)}$, where $\sigma_{(1)}$ and $\sigma_{(2)}$ are the two smallest elements of σ, and hence

$$f_0 = u \left[\frac{1}{\sigma_{(1)} + \sigma_{(2)}} - \frac{1}{\sigma_{(1)}} - \frac{1}{\sigma_{(2)}} \right].$$

Clearly, $f(\cdot)$ is a nondecreasing function. □

For the set $\mathcal{B}_u$, $\sigma_i = 2^{i-1}, \forall i$, implying that $f_0 = -7u/6$. Henceforth, let $g_0 := -f_0 = 7u/6$. Atamtürk and Narayanan [12], Proposition 1 tells us that the convex hull of the epigraph of a submodular function is completely described by its exponentially many polymatroid inequalities.

$$\text{conv}\{(x,z) \in \Re \times \{0,1\}^k \colon -x \geq f(S)\} = \{(x,z) \in \Re \times \{0,1\}^k \colon \pi z \leq -x - f_0, \forall \pi \in \text{ext}(EP_{f-f_0})\},$$

where $EP_{f-f_0} = \{\pi \in \Re^k \colon \pi(S) \leq f(S), \forall S \subseteq K\}$ is the extended polymatroid of f.

3.5.1 Separating a cut for $\text{conv}(\mathcal{B}_u)$

Observation 3.3. *Let $\mathbb{1}(t)$ denote the vector of binary coding of $t \in \mathbb{Z}_{++}$ and $a_t := \min\{a, u/t\}$. Then,*

$$\text{ext conv}(\mathcal{B}_u) = (0, \mathbf{0}, \mathbf{0}) \cup (a, \mathbf{0}, \mathbf{0}) \cup \bigcup_{t=1}^{b} \Big\{ (\mathbf{0}, \mathbb{1}(t), \mathbf{0}) \cup (a_t, \mathbb{1}(t), a_i \; \mathbb{1}(t)) \Big\}. \qquad (77)$$

Proof. Since $\mathcal{B}_u$ is a mixed $\{0,1\}$ set, its extreme points can be obtained by taking restrictions with respect to feasible values of z. The set of feasible values for z is the binary knapsack $\mathcal{K} = \{z \in \{0,1\}^k \colon \sum_{i=1}^k 2^{i-1} z_i \leq b\}$. Hence,

$$\operatorname{ext}\operatorname{conv}(\mathcal{B}_u) = \bigcup_{\hat{z} \in \mathcal{K}} \operatorname{ext}\left\{(x,z,v) \colon z = \hat{z}, x\left[\sum_{i=1}^k 2^{i-1}\hat{z}_i\right] \leq u, x \in [0,a], v_i = x\hat{z}_i, \forall i\right\}.$$

Since there is a bijection between the set of integers in $\{0,1,\ldots,b\}$ and $\mathcal{K}$, it follows that

$$\operatorname{ext}\operatorname{conv}(\mathcal{B}_u) = \bigcup_{t=0}^{b} \operatorname{ext}\{(x,z,v) \colon z = \hat{z}, tx \leq u, x \in [0,a], v_i = x, \forall i \colon \hat{z}_i = 1,\ v_i = 0, \forall i \colon \hat{z}_i = 0\}.$$

For $t = 0$, $x \in \{0, a\}$ at extreme points. For $t \geq 1$, $x \in \{0, a\}$ if $at \leq u$, otherwise $x \in \{0, u/t\}$. Thus, we get the proposed characterization of extreme points. □

The previous observation suggests that we can solve the following polar cut generating LP (CGLP) to compute a cut that separates (x^*, z^*, v^*) from $\operatorname{conv}(\mathcal{B}_u)$.

$$\begin{aligned} \max_{\substack{\alpha,\beta,\gamma,\alpha_0 \\ \alpha_0 \geq 0}} \quad & \alpha x^* + \beta^\top z^* + \gamma^\top v^* - \alpha_0 \\ \text{s.t.} \quad & a\alpha \leq \alpha_0 \\ & \beta^\top \mathbb{1}(t) \leq \alpha_0, \quad 1 \leq t \leq b \\ & a_t\alpha + \beta^\top \mathbb{1}(t) + a_t\gamma^\top \mathbb{1}(t) \leq \alpha_0, \quad 1 \leq t \leq b. \end{aligned} \tag{Polar}$$

Note that (Polar) has $2b$ constraints and hence is of pseudo polynomial size. One way to (slightly) reduce its size is to convexify a subset of $\operatorname{ext}(\mathcal{B}_u)$. Observe that if $y \in [0, \lfloor u/a \rfloor]$, then $x \in [0, a]$ at any feasible point of $\mathcal{B}_u$. Consider the subset

$$\left\{0 \leq x \leq a, \sum_{i=}^{k} 2^{i-1} z_i \leq \lfloor u/a \rfloor, v_i = xz_i, \forall i\right\},$$

whose convex hull can be derived using minimal covers from §3.3. Let $hx + Az + Bv \leq h_0$ be its convex hull. The reduced polar is then given by

$$
\begin{aligned}
\max_{\substack{\alpha,\beta,\gamma,\alpha_0 \\ \mu,\alpha_0 \geq 0}} \quad & \alpha x^* + \beta^\top z^* + \gamma^\top v^* - \alpha_0 \\
\text{s.t.} \quad & a\alpha \leq \alpha_0, \quad h^\top \mu \geq \alpha, \quad h_0^\top \mu = \alpha_0 \\
& A^\top \mu \geq \beta, \quad B^\top \mu \geq \gamma \\
& \beta^\top \mathbb{1}(t) \leq \alpha_0, \quad \lfloor u/a \rfloor + 1 \leq t \leq b \\
& \frac{u}{t}\alpha + \beta^\top \mathbb{1}(t) + \frac{u}{t}\gamma^\top \mathbb{1}(t) \leq \alpha_0, \quad \lfloor u/a \rfloor + 1 \leq t \leq b.
\end{aligned}
\tag{Polar2}
$$

If the ratio u/a is small, then (Polar2) is unlikely to provide a significant benefit over (Polar).

3.5.2 Disjunctive inequalities for $\mathrm{conv}(\mathcal{B}_u)$

In this subsection, we present valid inequalities to the convex hull of $\mathcal{B}_u$. Our main purpose is to strengthen and/or nontrivially aggregate two inequalities from §3.3 using the upper bound u on the product $x\sum_i 2^{i-1}z_i$. We derive these new inequalities using elementary variable disjunctions. In the set $\mathcal{X}_u$, y is an integer variable within $[0, b]$. Hence, one obvious disjunction is $\{y = 0\} \vee \{1 \leq y \leq b\}$. The other disjunction is on the continuous variable x : $\{0 \leq x \leq 7u/6\} \vee \{7u/6 \leq x \leq a\}$.

3.5.2.1 Disjunction on x

Case 1 : $a \geq 7u/6$**.** First, suppose that $a \geq 7u/6 = g_0$. Since $f(\cdot)$ is nondecreasing, $f_0 \leq f(S)$ for any $\emptyset \neq S \subseteq K$. Hence, the inequality $x \leq -f(S)$ is invalid for $x > -f_0 = g_0 = 7u/6$. We can reformulate $\mathcal{B}_u$ as the following disjunction after splitting the interval of x,

$$
\mathcal{B}_u = \left\{ \begin{array}{c} 0 \leq x \leq g_0 \\ -x \geq f(S) \\ \sum_{i=1}^k 2^{i-1} z_i \leq b \\ v_i = x z_i, z_i \in \{0,1\},\ \forall i \end{array} \right\} \bigcup \left\{ \begin{array}{c} g_0 \leq x \leq a \\ z = \mathbf{0}, v = \mathbf{0} \end{array} \right\}. \tag{78}
$$

where $g_0 = -f_0$. Consider a relaxation of the left side of the disjunction given by polymatroid inequalities and minimal covers from Corollary 3.3. This is indeed a relaxation because we are describing the convex hull of $-x \geq f(S), \forall S$ such that $\sigma(S) \leq b$, for which

polymatroids and covers may not be sufficient. Thus, we have

$$\operatorname{conv}(\mathcal{B}_u) \subseteq \operatorname{conv}\left(\mathcal{B}_u^L \bigcup \left\{ \begin{array}{c} g_0 \leq x \leq a \\ z = \mathbf{0}, v = \mathbf{0} \end{array} \right\}\right),$$

where $\mathcal{B}_u^L$ is the polyhedron

$$\mathcal{B}_u^L = \left\{ (x,z,v)\colon \begin{array}{ll} & v_j + \sum_{i\in I_j} v_i \leq |I_j| x, \quad \forall j \notin I \\ & g_0 z_j - v_j + \sum_{i\in I_j}(g_0 z_i - v_i) \leq |I_j|(g_0 - x), \quad \forall j \notin I \\ & \pi z + x \leq g_0, \quad \forall \pi \in \operatorname{ext}(EP_{f-f_0}) \\ & \max\{0, g_0 z_i + x - g_0\} \leq v_i \leq \min\{a z_i, x\}, \quad \forall i \\ & 0 \leq x \leq g_0 \end{array} \right\}$$

and, as before, I is the support of the binary coding of b and $I_j = \{i \in I\colon i > j\}$ for all $j \notin I$. Applying the Balas disjunctive result [16] yields the following relaxation for $\operatorname{conv}(\mathcal{B}_u)$.

$$\begin{aligned} v_j + \sum_{i\in I_j} v_i - |I_j| x^1 &\leq 0, \quad \forall j \notin I \\ g_0 z_j - v_j + \sum_{i\in I_j}(g_0 z_i - v_i) + |I_j| x^1 &\leq |I_j| g_0 \lambda, \quad \forall j \notin I \\ \pi z + x^1 &\leq g_0 \lambda, \quad \forall \pi \in \operatorname{ext}(EP_{f-f_0}) \\ v_i - g_0 z_i &\leq 0, \quad \forall i \\ v_i - x^1 &\leq 0, \quad \forall i \\ -v_i + g_0 z_i + x^1 &\leq g_0 \lambda, \quad \forall i \\ 0 \leq x^1 &\leq g_0 \lambda \\ g_0(1-\lambda) \leq x - x^1 &\leq a(1-\lambda) \\ 0 \leq \lambda \leq 1, x^1 &\geq 0, \quad z, v \geq \mathbf{0} \end{aligned} \tag{79}$$

After projecting out λ we get the following set of inequalities.

$$
\begin{aligned}
v_j + \sum_{i \in I_j} v_i &\leq |I_j| x^1, \quad j \notin I \\
-g_0 z_j + v_j + \sum_{i \in I_j} (-g_0 z_i + v_i) + |I_j| g_0 \left[1 - \frac{x}{a}\right] &\geq |I_j| \left[1 - \frac{g_0}{a}\right] x^1, \quad j \notin I \\
-\pi z + g_0 \left[1 - \frac{x}{a}\right] &\geq \left[1 - \frac{g_0}{a}\right] x^1, \quad \pi \in \text{ext}(EP_{f-f_0}) \\
v_i - g_0 z_i &\leq 0, \quad \forall i \\
v_i &\leq x^1, \quad \forall i \\
v_i - g_0 z_i + g_0 \left[1 - \frac{x}{a}\right] &\geq \left[1 - \frac{g_0}{a}\right] x^1, \quad \forall i \\
\frac{x}{a} + \frac{1}{g_0} \left[1 - \frac{g_0}{a}\right] x^1 &\leq 1
\end{aligned}
$$

Note that the last inequality is made redundant by the third inequality since $\pi \geq 0$ for every $\pi \in \text{ext}(EP_{f-f_0})$. After projecting out x^1 we get the following valid inequalities for $\text{conv}(\mathcal{B}_u)$. Define $\mu_0 := 1 - g_0/a$.

$$
\begin{aligned}
\frac{\mu_0}{|I_j|} \left[v_j + \sum_{i \in I_j} v_i \right] + g_0 z_i - v_i &\leq g_0 \left[1 - \frac{x}{a}\right], \quad j \notin I, \forall i && \text{(80a)} \\
\mu_0 v_{i'} + g_0 z_i - v_i &\leq g_0 \left[1 - \frac{x}{a}\right], \quad \forall i, i', i \neq i' && \text{(80b)}
\end{aligned}
$$

$$
\begin{aligned}
\frac{\mu_0}{|I_{j'}|} \left[v_{j'} + \sum_{i \in I_{j'}} v_i \right] + \frac{1}{|I_j|} \left[g_0 z_j - v_j + \sum_{i \in I_j} (g_0 z_i - v_i) \right] &\leq g_0 \left[1 - \frac{x}{a}\right], j, j' \notin I && \text{(81a)} \\
\mu_0 \, v_{i'} + \frac{1}{|I_j|} \left[g_0 z_j - v_j + \sum_{i \in I_j} (g_0 z_i - v_i) \right] &\leq g_0 \left[1 - \frac{x}{a}\right], j \notin I, \forall i' && \text{(81b)}
\end{aligned}
$$

$$
\begin{aligned}
\frac{\mu_0}{|I_j|} \left[v_j + \sum_{i \in I_j} v_i \right] + \pi z &\leq g_0 \left[1 - \frac{x}{a}\right], \quad \pi \in \text{ext}(EP_{f-f_0}), j \notin I && \text{(82a)} \\
\mu_0 \, v_i + \pi z &\leq g_0 \left[1 - \frac{x}{a}\right], \quad \pi \in \text{ext}(EP_{f-f_0}), i \in I && \text{(82b)} \\
v_i &\leq g_0 z_i, \quad \forall i. && \text{(82c)}
\end{aligned}
$$

Note that $j = j'$ for (81a) retrieves one of the original knapsack covers. Also (81a) cannot be obtained by aggregating the original covers. (82c) is a McCormick linearization

using the tighter bound g_0 instead of a. Separation over the above inequalities is easy. The extremal values of π are given by the sorting algorithm of Edmonds [38].

Proposition 3.12. *Inequalities* (80)–(82) *are valid to* $\text{conv}(\mathcal{B}_u)$ *assuming* $a \geq 7u/6$.

Case 2 : $a < 7u/6$**.** Finally, if $a < g_0$, then there is no need for a disjunction since the polymatroid inequalities will always be valid. In this case, a relaxation for $\text{conv}(\mathcal{B}_u)$ is

$$\begin{aligned} \pi z + x &\leq g_0, \quad \pi \in \text{ext}(EP_{f-f_0}) && \text{(83a)} \\ v_j + \sum_{i \in I_j} v_i &\leq |I_j| x, \quad j \notin I && \text{(83b)} \\ az_j - v_j + \sum_{i \in I_j} (az_i - v_i) &\leq |I_j|(a - x), \quad j \notin I. && \text{(83c)} \end{aligned}$$

3.5.2.2 Disjunction on y

Here we do a disjunction on $y = 0 \vee 1 \leq y \leq b$. Let $a^\perp = \min\{a, u\}$. Note that by definition, $a^\perp \leq g_0 = 7u/6$. $\mathcal{B}_u$ can be written as the following disjunction.

$$\mathcal{B}_u = \left\{ \begin{array}{c} 0 \leq x \leq a^\perp \\ -x \geq f(S) \\ 1 \leq \sum_{i=1}^k 2^{i-1} z_i \leq b \\ v_i = x z_i, z_i \in \{0,1\}, \ \forall i \end{array} \right\} \bigcup \left\{ \begin{array}{c} 0 \leq x \leq a \\ z = \mathbf{0}, v = \mathbf{0} \end{array} \right\}. \tag{84}$$

Assuming $a \geq g_0$, note that the disjunctive system in (84) is quite similar to the one in (78), with the following differences in (84).

1. The left side of disjunction has a tighter upper bound on x : $x \leq a^\perp \leq g_0$.
2. The left side of disjunction has a lower bound on $\sum_i 2^{i-1} z_i$.
3. The right side of disjunction has a weaker lower bound on x : $x \geq 0 < g_0$.

Hence, we can setup the disjunctive system as in (79) but this time with the modified bound $a^{\perp}$.

$$
\begin{aligned}
v_j + \sum_{i \in I_j} v_i - |I_j| x^1 &\leq 0, \quad \forall j \notin I \\
a^{\perp} z_j - v_j + \sum_{i \in I_j} (a^{\perp} z_i - v_i) + |I_j| x^1 &\leq |I_j| a^{\perp} \lambda, \quad \forall j \notin I \\
\pi z + x^1 &\leq g_0 \lambda, \quad \forall \pi \in \text{ext}(EP_{f-f_0}) \\
v_i - a^{\perp} z_i &\leq 0, \quad \forall i \\
v_i - x^1 &\leq 0, \quad \forall i \\
-v_i + a^{\perp} z_i + x^1 &\leq a^{\perp} \lambda, \quad \forall i \\
0 \leq x^1 &\leq a^{\perp} \lambda \\
0 \leq x - x^1 &\leq a(1 - \lambda) \\
0 \leq \lambda \leq 1, x^1 \geq 0, &\quad z, v \geq \mathbf{0}
\end{aligned}
$$

After projecting out λ we get the following set of inequalities. Define $\tau_0 := 1 - a^{\perp}/a$.

$$
\begin{aligned}
v_j + \sum_{i \in I_j} v_i &\leq |I_j| x^1, \quad j \notin I \\
-a^{\perp} z_j + v_j + \sum_{i \in I_j} (-a^{\perp} z_i + v_i) + |I_j| a^{\perp} \left[1 - \frac{x}{a}\right] &\geq |I_j| \tau_0 x^1, \quad j \notin I \\
-\pi z + g_0 \left[1 - \frac{x}{a}\right] &\geq \left[1 - \frac{g_0}{a}\right] x^1, \quad \pi \in \text{ext}(EP_{f-f_0}) \\
v_i - a^{\perp} z_i &\leq 0, \quad \forall i \\
v_i &\leq x^1, \quad \forall i \\
v_i - a^{\perp} z_i + a^{\perp} \left[1 - \frac{x}{a}\right] &\geq \tau_0 x^1, \quad \forall i \\
\frac{x}{a} + \frac{\tau_0}{a^{\perp}} x^1 &\leq 1
\end{aligned}
$$

After projecting out x^1 we get the following valid inequalities for $\text{conv}(\mathcal{B}_u)$.

Proposition 3.13. *The following inequalities are valid to* $\text{conv}(\mathcal{B}_u)$.

$$
\frac{\tau_0}{|I_j|} \left[v_j + \sum_{i \in I_j} v_i \right] + a^{\perp} z_i - v_i \leq a^{\perp} \left[1 - \frac{x}{a}\right], \quad j \notin I, \forall i' \tag{85a}
$$

$$
\tau_0 v_{i'} + a^{\perp} z_i - v_i \leq a^{\perp} \left[1 - \frac{x}{a}\right], \quad \forall i, i', i \neq i' \tag{85b}
$$

$$\frac{\tau_0}{|I_{j'}|}\left[v_{j'}+\sum_{i\in I_{j'}} v_i\right]+\frac{1}{|I_j|}\left[a^\perp z_j - v_j + \sum_{i\in I_j}(a^\perp z_i - v_i)\right] \le a^\perp\left[1-\frac{x}{a}\right], j, j' \notin I \quad (86a)$$

$$\tau_0 v_{i'} + \frac{1}{|I_j|}\left[a^\perp z_j - v_j + \sum_{i\in I_j}(a^\perp z_i - v_i)\right] \le a^\perp\left[1-\frac{x}{a}\right], j \notin I, \forall i' \quad (86b)$$

$$\frac{\mu_0}{|I_j|}\left[v_j+\sum_{i\in I_j} v_i\right]+\pi z \le g_0\left[1-\frac{x}{a}\right], \quad \pi \in \text{ext}(EP_{f-f_0}), j \notin I \quad (82a)$$

$$\mu_0 v_i + \pi z \le g_0\left[1-\frac{x}{a}\right], \quad \pi \in \text{ext}(EP_{f-f_0}), i \in I \quad (82b)$$

$$v_i \le a^\perp z_i, \quad \forall i. \quad (87a)$$

We note that many of the inequalities of Proposition 3.13 are similar in structure to those of Proposition 3.12. For example, (80) and (85) are similar, though neither seems to dominate the other, since $a^\perp \le g_0$ and $\tau_0 \ge \mu_0$. Similarly for (81) and (86). Inequality (87a) is a stronger McCormick than (82c). We also note that disjunction on y produces more inequalities by aggregating two polymatroids or by aggregating one polymatroid and one minimal cover.

3.5.3 Computational results

The inequalities of Propositions 3.12 and 3.13 were implemented on the product bundling instances from §3.4.2.2. Recall that these problems are formulated as

$$\max\left\{\sum_{c\in C}\sum_{p\in P} x_p y_c \colon x_p y_c \le D_{cp},\ x_p, y_c \in \mathbb{Z}_+ \forall c, p\right\}.$$

Hence, explicit upper bounds are readily available for the bilinear terms. Since $x_p \in \mathbb{Z}$, we replace g_0 with $\lfloor g_0 \rfloor$ and $\lceil g_0 \rceil$ on the left and right disjunction, respectively, while performing disjunction on x_p in (78). Upper bounds on the individual variables x_p and y_c are obtained from Proposition 3.10.

Since there is an exponential family of valid inequalities, the following strategy was used to control the number of cuts generated : we only checked separation over inequalities of the type (80), (81), (82), (85), and (86). The cut separator was called only at nodes of

depth no more than 20 and at each such node, at most 3 rounds of separation were allowed. During any round of separation, for each integer variable y_c, at most one cut of each type was added (one of (80) or (85), one of (81) or (86), one of (82)). The separated cut was the one that produced the maximum violation across all covers and polymatroids for all bilinear terms involving y_c. The stronger McCormick (87a) was added to the problem reformulation (B-MIBLP).

Results of our experiment are presented in Tables 14, 15, and 16. Columns `Couenne`, (M-MIBLP), and (B-MIBLP) are reproduced from §3.4.2.2 for comparison purposes. For `watts` instances, we present optimality gap at termination after one hour if not solved to optimality. We also implemented the (Polar) CGLP with standard ℓ_∞ normalization constraint. However, we found no benefit with this cut separator. They were neither more helpful in closing the root gap nor were they allowing any speed-up of the solution process. In fact, the polar CGLP approach was processing more nodes and yielding poorer optimality gaps than our proposed inequalities. Hence, we do not report its performance numbers.

Table 14: Cuts from bounded bilinear term for 27 random instances of product bundling with $|C| = 10, |P| = 30$. Columns `Couenne`, (M-MIBLP), and (B-MIBLP) are reproduced from §3.4.2.2.

	`Couenne`	(M-MIBLP)	(B-MIBLP) + Covers + Polymatroids	(B-MIBLP)
Average Nds	388824	735621	186170	176372
Average T (sec.)	2103.17	2409.01	2343.10	2339.65
Average Cuts	–	–	1126	–
Average % Rgp-cl	–	–	0.1	–
(a) # solved	13	9	10	8
(b) # fastest optimal	10	4	0	0
(c) Average time gain (sec.)	623.90	11.00	0.00	0.00
(d) # best feasible	1	6	13	10
(e) Average $\mu_{\mathcal{I}}$	98.94	20.43	88.34	85.41
(f) Average $\omega_{\mathcal{I}}$	13.71	4.98	0.07	0.29

We observe from the above tables that a large number of proposed cuts are being added during branch-and-bound. This may sometimes cause the branch-and-bound search to

Table 15: Cuts from bounded bilinear term for 27 random instances of product bundling with $|C| = 20, |P| = 50$. Columns `Couenne`, (M-MIBLP), and (B-MIBLP) are reproduced from §3.4.2.2.

	`Couenne`	(M-MIBLP)	(B-MIBLP) + Covers + Polymatroids	(B-MIBLP)
Average Nds	99790	241450	126814	82620
Average T (sec.)	2776.51	3600.00	3182.10	3309.07
Average Cuts	–	–	1626	–
Average % Rgp-cl	–	–	0.1	–
(a) # solved	8	0	5	5
(b) # fastest optimal	8	0	0	0
(c) Average time gain (sec.)	1342.83	0.00	0.00	0.00
(d) # best feasible	0	1	13	9
(e) Average $\mu_{\mathcal{I}}$	2290.93	343.71	452.18	479.47
(f) Average $\omega_{\mathcal{I}}$	53.94	24.36	2.29	2.87

Table 16: Cuts from bounded bilinear term for the `watts` instances of product bundling. Columns `Couenne`, (M-MIBLP), and (B-MIBLP) are reproduced from §3.4.2.2.

Instance	`Couenne`		M-MIBLP		B-MIBLP + Covers + Polymatroids				B-MIBLP	
	Nds	T	Nds	T	Nds	T	Cuts	Rgp-cl	Nds	T
`5x41`	305800	40%	1217175	165%	36571	285	1547	0	33762	176
`5x41m`	1044023	6%	1301411	295%	21073	161	555	0	16426	226
`9x60`	120260	461%	319347	277%	153988	124%	4277	0	86562	65%
`10x60`	97180	338%	239071	252%	120839	71%	6418	0	69911	59%
`10x60d`	112030	192%	291302	157%	35807	426	942	0	60945	1902

slowdown. The cuts did not help close any root gap. But these cuts can be potentially useful since they were allowing `Cplex` to produce better quality solutions. We found that one possible reason for this phenomenon could be that more integer feasible solutions were occurring from node LPs as opposed to heuristic search. Also, addition of cuts increased the number of problems solved to optimality by `Cplex` in one hour.

3.6 General expansion knapsack

Suppose that instead of using binary expansion for a general integer, we want to use α-nary representation for some $\alpha \in \mathbb{Z}_{++}, \alpha \geq 2$. Then the discrete set $\mathcal{K}(\alpha, b)$ in the z-space is a special case of the multi-item knapsack with divisible coefficients (*sequential knapsack polytope*) given by

$$\mathcal{K}(\alpha, b) := \left\{ z \in \mathbb{Z}_+^n \colon \sum_{i=1}^{n} \alpha^{i-1} z_i \leq b,\, 0 \leq z_i \leq \alpha - 1, \forall i \right\}, \tag{88}$$

where $\alpha \in \mathbb{Z}_{++}, \alpha \geq 2$. Let $N = \{1, \ldots, n\}$. Since the knapsack weights are positive integers, we also assume that $b \in \mathbb{Z}_{++}$. Since both α and b are natural numbers, we can write down a unique base α representation for b. For each item i in the knapsack, associate a integer coefficient θ_i that denotes the contribution of item i in filling the knapsack capacity b. In particular, we have

$$\begin{aligned} b &= \sum_{i \in N} \theta_i \alpha^{i-1} \\ &= \sum_{i \in I} \theta_i \alpha^{i-1}, \end{aligned} \tag{89}$$

where $I := \{i \in N \colon \theta_i \geq 1\}$ is the set of items that make a positive contribution towards b. We refer to $(I, \boldsymbol{\theta})$ as the *packing* of the knapsack capacity b. Since $\boldsymbol{\theta} \in \mathcal{K}(\alpha, b)$, we must have $\theta_i \leq \alpha - 1$ for all $i \in N$. A few simple observations about $\mathcal{K}(\alpha, b)$ are given below.

Observation 3.4. $\operatorname{conv}(\mathcal{K}(\alpha, b))$ *is full-dimensional if and only if* $n \leq \lfloor \log_\alpha b \rfloor + 1$. *If* n *is strictly less than* $\lfloor \log_\alpha b \rfloor + 1$, *then the knapsack constraint is redundant and we get* $\operatorname{conv}(\mathcal{K}(\alpha, b)) = [0, \alpha - 1]^n$. *Henceforth, we assume that* $n = \lfloor \log_\alpha b \rfloor + 1$.

Observation 3.5. *The base* α *representation of* b *must use* α^{n-1}. *In other words,* $n \in I$.

Observation 3.6. *The upper bound on the* n^{th} *item is* θ_n, *i.e.* $z_n \leq \theta_n$ *is a valid inequality for* $\operatorname{conv}(\mathcal{K}(\alpha, b))$.

Observation 3.7. *There is a bijection between* $\mathcal{K}(\alpha, b)$ *and the set of integers in the interval* $[0, b]$.

Observation 3.8. *For any* $2 \leq k \leq n$, $(\alpha - 1) \sum_{i=1}^{k} \alpha^{i-1} = \alpha^k - 1 < \alpha^k$.

The main result of this section is the following family of inequalities.

Proposition 3.14. *For any $j \in N$, define $I_j := \{i \in I \colon i > j\} = \{i_1, \ldots, i_{r_j}, i_{r_j+1} := n\}$ (assumed to be sorted in increasing order). Then the inequality*

$$z_j + \sum_{i \in I_j} \pi_i (z_i - \theta_i) \leq \theta_j \tag{90}$$

is valid to $\mathrm{conv}(\mathcal{K}(\alpha, b))$*, for all $j \in N$, where the coefficients π are given by*

$$\begin{array}{rcl} \pi_{i_1} & = & \alpha - 1 - \theta_j \\ \pi_{i_t} & = & (\alpha - \theta_{i_{t-1}})\pi_{i_{t-1}}, \quad \forall 2 \leq t \leq r_j + 1. \end{array} \tag{91}$$

Since the coefficient of the first variable is $\pi_{i_1} = \alpha - 1 - \theta_j$ and the subsequent coefficients are recursively obtained as $\pi_{i_t} = (\alpha - \theta_{i_{t-1}})\pi_{i_{t-1}}$, it is noted that if $\theta_j = \alpha - 1$ then the valid inequality in (90) reduces to an upper bound $z_j \leq \alpha - 1$. The inequality corresponding to the last item, i.e. $j = n$, is also a upper bound $z_n \leq \theta_n$, which we already observed to be valid. The next observation is easy to verify.

Observation 3.9. *In the special case $\alpha = 2$, the knapsack set $\mathcal{K}(2, b) = \mathcal{K}$, cf.* (56)*, and the inequalities* (90) *correspond to minimal covers of $\mathcal{K}$, for $j \notin I$, and the trivial facets $z_j \leq 1$, for $j \in I$.*

We demonstrate the proposed inequalities with an example.

Example 3.2. Let $\alpha = 5$ and $b = 8320$ so that $n = 6$, $I = \{2, 3, 4, 5, 6\}$, and $\theta_2 = 4, \theta_3 = 2, \theta_4 = 1, \theta_5 = 3, \theta_6 = 2$. The convex hull as given by the PORTA software of Christof and Löbel, is

$$\begin{array}{rcll} \mathrm{conv}(\mathcal{K}(5, 8320)) & = & \Big\{ z \in \Re^6_+ \colon & z_i \geq 0, \quad i = 1, \ldots, 6 \\ & & & z_i \leq 4, \quad i = 1, \ldots, 5 \\ & & & z_6 \leq 2 \\ & & & z_5 + z_6 \leq 5 \\ & & & z_4 + 3z_5 + 6z_6 \leq 22 \\ & & & z_3 + 2z_4 + 8z_5 + 16z_6 \leq 60 \\ & & & z_1 + 4z_2 + 4z_3 + 12z_4 + 48z_5 + 96z_6 \leq 372 \Big\}. \end{array}$$

One can verify that the nontrivial facets of this convex hull are the inequalities from (90) for $j = 6, 5, 4, 3, 1$, respectively. Since $\theta_2 = 4 = \alpha - 1$, π_{i_1} and hence $\pi_{i_t}, \forall t$, is equal to zero. Hence inequality (90) for $j = 2$ simply becomes $z_2 \leq \theta_2 = 4$. □

3.6.1 Proof of validity

We will give a lifting argument to prove the validity of the proposed inequalities in (90). For every $j \in N$, define the sets $I_j := \{i \in I \colon i > j\}$ and $I_j^- := \{i \in I \colon i < j\}$. Consider the lower dimensional set

$$\mathcal{K}(\alpha, b)_j := \{z \in \mathcal{K}(\alpha, b) \colon z_i = \theta_i,\ \forall i \in I_j\}\,. \tag{92}$$

Lemma 3.2. *For any $j \in N$, the inequality $z_j \leq \theta_j$ is valid for* $\mathrm{conv}(\mathcal{K}(\alpha, b)_j)$ *and represents a lower-dimensional face if and only if $j \in I$.*

Proof. Consider the set $\mathcal{K}(\alpha, b)_j$. For any z that belongs to this set, we have

$$\begin{aligned}\sum_{i=1}^{j} \alpha^{i-1} z_i + \sum_{i>j, i\notin I} \alpha^{i-1} z_i &\leq b - \sum_{i \in I_j} \alpha^{i-1}\theta_i, \\ &= \sum_{i \in I_j^-} \alpha^{i-1}\theta_i + \alpha^{j-1}\theta_j.\end{aligned}$$

Since $\theta_i \leq \alpha - 1$, for all i, and $\sum_{i=1}^{j} \alpha^{i-1}(\alpha - 1) = \alpha^j - 1 < \alpha^j$, it must be that $z_i = 0$ for all $i > j, i \notin I$. Thus, the above inequality is in fact

$$\sum_{i=1}^{j} \alpha^{i-1} z_i \leq \sum_{i \in I_j^-} \alpha^{i-1}\theta_i + \alpha^{j-1}\theta_j \tag{93}$$

Now, suppose that $z_j = \theta_j + \kappa$, for some integer $\kappa \geq 1$. Then the left hand side becomes

$$\begin{aligned}\sum_{i=1}^{j} \alpha^{i-1} z_i &\geq (\theta_j + \kappa)\alpha^{j-1} \\ &= \theta_j \alpha^{j-1} + \kappa\alpha^{j-1} \\ &> \theta_j \alpha^{j-1} + \sum_{i \in I_j^-} \alpha^{i-1}(\alpha - 1) \\ &\geq \theta_j \alpha^{j-1} + \sum_{i \in I_j^-} \alpha^{i-1}\theta_i\end{aligned}$$

and hence a contradiction to $z \in \mathcal{K}(\alpha, b)_j$. This proves the validity of $z_j \leq \theta_j$ for $\mathrm{conv}(\mathcal{K}(\alpha, b)_j)$.

For any $z \in \mathcal{K}(\alpha, b)_j$, since $z_i = 0$ for all $i > j, i \notin I$, and $z_i = \theta_i$, for all $i \in I_j$, it must be that dimn $\text{conv}(\mathcal{K}(\alpha, b)_j) \leq j$. Consider two cases. First, suppose that $j \in I$. Then the unit vectors $\mathbf{e}_1, \ldots, \mathbf{e}_j$ and $\mathbf{0}$ belong to $\text{conv}(\mathcal{K}(\alpha, b)_j)$ and hence dimn $\text{conv}(\mathcal{K}(\alpha, b)_j) = j$. Now, if $j \notin I$, then it must be that $z_i = 0$, for all $i > i_j^* := \max\{i_t \colon i_t \in I_j^-\}$. Note that $i_j^* = 0$ if I_j^- is empty. Then the unit vectors $\mathbf{e}_1, \ldots, \mathbf{e}_{i_j^*}$ and $\mathbf{0}$ belong to $\text{conv}(\mathcal{K}(\alpha, b)_j)$ and hence, dimn $\text{conv}(\mathcal{K}(\alpha, b)_j) = i_j^*$ for this case.

Consider the face $\mathcal{F}_j := \{z \in \text{conv}(\mathcal{K}(\alpha, b)_j) \colon z_j = \theta_j\}$. Then for all $z \in \mathcal{F}_j$ we have from (93) that $\sum_{i=1}^{j-1} \alpha^{i-1} z_i \leq \sum_{i \in I_j^-} \alpha^{i-1} \theta_i$. Hence, $z_i = 0$, for all i such that $i_j^* < i \leq j-1$. This implies that $\text{dimn}(\mathcal{F}_j) = i_j^*$. Thus, if $j \in N \setminus I$, then $\mathcal{F}_j$ is not a proper face since $\mathcal{F}_j = \text{conv}(\mathcal{K}(\alpha, b)_j)$. □

The proposed valid inequality (90) is obtained by lifting the upper bound $z_j \leq \theta_j$ in the variables $z_i, i \in I_j$. Let $I_j = \{i \in I \colon i > j\} = \{i_1, i_2, \ldots, i_{r_j}, n\}$ for some $r_j \geq 0$. Lifting is performed sequentially in the variable order $z_{i_1}, z_{i_2}, \ldots, z_{i_{r_j}}, z_n$. For convenience, $i_{r_j+1} := n$.

Proof of Proposition 3.14. Consider the lifting procedure for the first variable in the sequence z_{i_1}. The goal is to show that $z_j + (\alpha - 1 - \theta_j)(z_{i_1} - \theta_{i_1}) \leq \theta_j$ is a valid inequality to $\mathcal{K}(\alpha, b)_{j \cup \{i_1\}}$, where

$$\mathcal{K}(\alpha, b)_{j \cup \{i_1\}} = \Big\{ z \colon \sum_{i=1}^{j} \alpha^{i-1} z_i + \alpha^{i_1 - 1}(z_{i_1} - \theta_{i_1}) + \sum_{i > j, i \notin I} \alpha^{i-1} z_i \leq \sum_{i \in I_j^-} \alpha^{i-1} \theta_i + \alpha^{j-1} \theta_j$$
$$z_l \in [0, \alpha - 1] \cap \mathbb{Z}_+,\ l \in N \setminus \{i_2, \ldots, i_{r_j}, n\} \Big\}.$$

Let $z_j + \pi_{i_1}(z_{i_1} - \theta_{i_1}) \leq \theta_j$ be the lifted inequality and define $\delta_{i_1} := z_{i_1} - \theta_{i_1}$, for $z_{i_1} \in [0, \alpha - 1] \cap \mathbb{Z}_+ \setminus \theta_{i_1}$. Using the definition in (92), it follows that $\mathcal{K}(\alpha, b)_{j \cup \{i_1\}} = \mathcal{K}(\alpha, b)_{i_1}$. Then, Lemma 3.2 applied to $\mathcal{K}(\alpha, b)_{i_1}$ implies that δ_{i_1} must be negative.

The lifting function is given by $\psi(\delta_{i_1}) := \min\{\theta_j - z_j : z \in \mathcal{K}(\alpha, b)_{i_1}, z_{i_1} - \theta_{i_1} = \delta_{i_1}\}$. Since $\delta_{i_1} < 0$, it is easily seen that the optimal solution for this minimization problem is achieved at $z_j = \alpha - 1$. Hence, $\psi(\delta_{i_1}) = \theta_j - a + 1$. Thus, the lifted inequality is valid for $\mathcal{K}(\alpha, b)_{i_1}$ if $\pi_{i_1} \delta_{i_1} \leq \psi(\delta_{i_1})$ for all $\delta_{i_1} \in \{-\theta_{i_1}, -\theta_{i_1} + 1, \ldots, -1\}$. After rearranging the terms, we see that the lifted inequality is valid if we choose $\pi_{i_1} = a - 1 - \theta_j$.

The lifting coefficients of the remaining variables in the lifting sequence are proved by induction. The base case is lifting the next variable z_{i_2}. Let

$$z_j + (\alpha - 1 - \theta_j)(z_{i_1} - \theta_{i_1}) + \pi_{i_2}(z_{i_2} - \theta_{i_2}) \leq \theta_j$$

be the lifted inequality that is required to be valid for $\mathcal{K}(\alpha, b)_{j \cup \{i_1, i_2\}} = \mathcal{K}(\alpha, b)_{i_2}$. Define $\delta_{i_2} := z_{i_2} - \theta_{i_2}$, for $z_{i_2} \in [0, \alpha - 1] \cap \mathbb{Z}_+ \setminus \theta_{i_2}$. The lifting function is

$$\psi(\delta_{i_2}) := \min\{\theta_j - z_j - \pi_{i_1}(z_{i_1} - \theta_{i_1}) \colon z \in \mathcal{K}(\alpha, b)_{i_2}, z_{i_2} - \theta_{i_2} = \delta_{i_2}\}.$$

Since $(\alpha - 1)(\alpha^{j-1} + \alpha^{i_1 - 1}) < \alpha^{i_2 - 1}$ and δ_{i_2} is negative due to Lemma 3.2, the optimal solution is attained at $z_j = z_{i_1} = \alpha - 1$, all others zero. Hence, the lifting function is $\psi(\delta_{i_2}) = \theta_j + \pi_{i_1}\theta_{i_1} - (\alpha - 1)(\pi_{i_1} + 1)$. After simplifying the expression we get $\psi(\delta_{i_2}) = \pi_{i_1}(\theta_{i_1} - \alpha)$. Then, a sufficient condition for the lifted inequality to be valid is $\pi_{i_2}\delta_{i_2} \leq \pi_{i_1}(\theta_{i_1} - \alpha)$, for all $\delta_{i_2} \in \{-\theta_{i_2}, -\theta_{i_2} + 1, \ldots, -1\}$. This gives us that $\pi_{i_2} = \pi_{i_1}(\alpha - \theta_{i_1})$ for maximal lifting.

Now, consider lifting the variable $z_{i_{t+1}}$, for some $t \geq 2$, and by induction hypothesis, assume that $\pi_{i_k} = (a - \theta_{i_{k-1}})\pi_{i_{k-1}}$, for $k \leq t$. Following the same procedure as before, we get the lifting function as

$$\begin{aligned} \psi(\delta_{i_{t+1}}) \ := \quad & \min \quad \theta_j - z_j - \textstyle\sum_{l=1}^{t} \pi_{i_l}(z_{i_l} - \theta_{i_l}) \\ & \text{s.t.} \quad z \in \mathcal{K}(\alpha, b)_{i_{t+1}} \\ & \qquad z_{i_{t+1}} - \theta_{i_{t+1}} = \delta_{i_{t+1}}. \end{aligned}$$

The optimal solution to this minimization problem is attained at $z_j = z_{i_1} = \cdots = z_{i_t} = \alpha - 1$, all other variables set to zero. Hence, the lifting function evaluates to

$$\begin{aligned} \psi(\delta_{i_{t+1}}) &= -\Big[\pi_{i_1} + \textstyle\sum_{l=1}^{t} \pi_{i_l}(\alpha - 1 - \theta_{i_l})\Big] \\ &= -\Big[\pi_{i_1}(a - \theta_{i_1}) - \pi_{i_2} \ + \ \pi_{i_2}(a - \theta_{i_2}) - \pi_{i_3} \ + \ldots \\ &\qquad\qquad + \pi_{i_{t-1}}(a - \theta_{i_{t-1}}) - \pi_{i_t} \ + \ \pi_{i_t}(a - \theta_{i_t})\Big] \\ &= -\pi_{i_t}(a - \theta_{i_t}) \end{aligned}$$

Finally, maximal lifting is guaranteed by setting $\pi_{i_{t+1}} = \pi_{i_t}(a - \theta_{i_t})$. This completes the induction process and our proof. □

3.6.2 Facets of $\text{conv}(\mathcal{K}(\alpha, b))$

We next show that the valid inequalities of Proposition 3.14 are facets of $\text{conv}(\mathcal{K}(\alpha, b))$. We will use first principles to prove this facial property by enumerating sufficient number of affinely independent points that belong to the face induced by this inequality.

Proposition 3.15. *Inequalities* (90) *are facet-defining to* $\text{conv}(\mathcal{K}(\alpha, b))$ *for all* $j \in N$.

Proof. Consider the inequality (90) for some $j \in N$.

$$\begin{aligned} z_j + (\alpha - 1 - \theta_j)(z_{i_1} - \theta_{i_1}) + \sum_{t=2}^{r_j+1} (\alpha - \theta_{i_t-1})\pi_{i_t-1}(z_{i_t} - \theta_{i_t}) &\leq \theta_j \\ \Rightarrow z_j + (\alpha - 1 - \theta_j)(z_{i_1} - \theta_{i_1}) + (\alpha - 1 - \theta_j) \sum_{t=2}^{r_j+1} \left[\prod_{k=1}^{t-1} (\alpha - \theta_{i_k}) \right] (z_{i_t} - \theta_{i_t}) &\leq \theta_j \end{aligned}$$

As before we denote $I_j = \{i_1, \ldots, i_{r_j}, i_{r_j+1} := n\}$.

Consider a point z fixed as : $z_j = z_{i_1} = \cdots = z_{i_{r_j}} = \alpha - 1$, $z_n = \theta_n - 1$, and $z_k = 0$, for $k \in N \setminus (j \cup I_j)$. This point belongs to $\mathcal{K}(\alpha, b)$ because

$$\begin{aligned} b &= \sum_{i \in I} \theta_i \alpha^{i-1} \\ &\geq \alpha^{n-1} + (\theta_n - 1)\alpha^{n-1} + \sum_{i \in I_j} \theta_i \alpha^{i-1} + \theta_j \alpha^{j-1} \\ &> \sum_{i \in I_j} (\alpha - 1 - \theta_i)\alpha^{i-1} + (\alpha - 1 - \theta_j)\alpha^{j-1} + (\theta_n - 1)\alpha^{n-1} + \sum_{i \in I_j} \theta_i \alpha^{i-1} + \theta_j \alpha^{j-1} \\ &= (\theta_n - 1)\alpha^{n-1} + \sum_{i \in I_j} (\alpha - 1)\alpha^{i-1} + (\alpha - 1)\alpha^{j-1} \end{aligned}$$

where the strict inequality is from Observation 3.8.

Now we claim that this point also belongs to the face induced by (90). Indeed,

$$\begin{aligned}
\alpha - 1 + \quad & (\alpha - 1 - \theta_j)(\alpha - 1 - \theta_{i_1}) + (\alpha - 1 - \theta_j) \sum_{t=2}^{r_j} (\alpha - 1 - \theta_{i_t}) \prod_{k=1}^{t-1} (\alpha - \theta_k) \\
= \quad & \alpha - 1 \; + \; (\alpha - 1 - \theta_j) \Bigg[\alpha - 1 - \theta_{i_1} + (\alpha - \theta_{i_1})(\alpha - 1 - \theta_{i_2}) \\
& \qquad + \sum_{t=3}^{r_j} (\alpha - 1 - \theta_{i_t}) \prod_{k=1}^{t-1} (\alpha - \theta_{i_k}) \; + \; \prod_{k=1}^{r_j} (\alpha - \theta_{i_k})(-1) \Bigg] \\
= \quad & \alpha - 1 \; + \; (\alpha - 1 - \theta_j) \Bigg[-1 + (\alpha - \theta_{i_1})(\alpha - \theta_{i_2}) \\
& \qquad + \sum_{t=3}^{r_j} (\alpha - 1 - \theta_{i_t}) \prod_{k=1}^{t-1} (\alpha - \theta_{i_k}) \; + \; \prod_{k=1}^{r_j} (\alpha - \theta_{i_k})(-1) \Bigg] \\
= \quad & \alpha - 1 \; + \; (\alpha - 1 - \theta_j) \Bigg[-1 \; + \; \prod_{k=1}^{r_j} (\alpha - \theta_{i_k}) \; + \; \prod_{k=1}^{r_j} (\alpha - \theta_{i_k})(-1) \Bigg] \\
= \quad & \alpha - 1 \; - \; (\alpha - 1 - \theta_j) \\
= \quad & \theta_j.
\end{aligned}$$

Following similar steps as above, we can show that for any $1 \leq l \leq r_j$, the point : $z_j = z_{i_1} = \cdots = z_{i_{l-1}} = \alpha - 1$, $z_{i_l} = \theta_{i_l} - 1$, $z_{i_t} = \theta_{i_t}$, for all $t \in \{l+1, \ldots, r_j + 1\}$, and $z_k = 0$, for all $k \in N \setminus (j \cup I_j)$, belongs to $\mathcal{K}(\alpha, b)$ and to the face induced by (90).

Now consider the following $n + 1$ points.

1. Choose an $l \in \{1, \ldots, r_j + 1\}$. Denote $i_0 := j$. Set $z_j = z_{i_1} = \cdots = z_{i_{l-1}} = \alpha - 1$, $z_{i_l} = \theta_{i_l} - 1$, $z_{i_t} = \theta_{i_t}$, for all $t \in \{l+1, \ldots, r_j + 1\}$, and

   (a) $z_k = 0$, for all $k \in N \setminus (j \cup I_j)$, or

   (b) $z_k = 1$, for some k such that $i_{l-1} < k < i_l$, and $z_k = 0$, otherwise. Since $\alpha^{k-1} < \alpha^{i_l - 1}$, this point is in $\mathcal{K}(\alpha, b)$.

2. Set $z_j = \alpha - 1, z_{i_1} = \theta_{i_1} - 1$, $z_{i_t} = \theta_{i_t}$, for all $t \in [2, r_j + 1]$, $z_l = 1$ for some $1 \leq l < j$, and $z_k = 0$, otherwise.

3. $z_i = \theta_i, \forall i \in j \cup I_j$, and $z_k = 0, \forall k \in N \setminus (j \cup I_j)$.

All the above points belong to the face induced by (90) and are affinely independent. $\square$

For $\alpha = 2$, we already proved in Proposition 3.9 that (90), which reduces to (66), and variable bounds describe the convex hull of $\mathcal{K}(2, b)$. Based on PORTA [32] experiments, we find that inequalities (90) along with variable bounds seem to be sufficient to characterize the convex hull of $\mathcal{K}(\alpha, b)$, for all integers $\alpha \geq 2$ and all integers b.

Conjecture 3.1. *The sequentially lifted inequalities* (90) *along with variable bounds are sufficient to define* $\text{conv}(\mathcal{K}(\alpha, b))$.

3.7 Conclusion

In this study, we presented a MILP reformulation (B-MIBLP) for the mixed integer bilinear problem (MIBLP). The idea behind constructing the reformulation was to use binary expansion of general integer variables. We investigated this reformulation by conducting a polyhedral study in the extended space. The set of interest turned out to be a special case of the sequential knapsack polytope. A polynomial size description was provided for the convex hull of this set using a previous result on minimal covers of superincreasing knapsacks. We implemented our cuts on five sets of instances and compared their performance against (i) a MINLP solver for problem (MIBLP), and (ii) a branching scheme within a MILP solver for relaxation (M-MIBLP).

Our experiments suggest that the cuts were more effective for test instances with a bilinear objective function and linear constraints. Even if our cuts were not always successful in closing a significant amount of the root gap on general bilinear problems, they often helped branch-and-cut search deeper down the tree. The results lend credence to our primary motivation for this study: that on certain class of problems, adopting a MILP solution procedure for solving mixed integer bilinear problems can be beneficial. Finally, we emphasize that the cuts derived in this study are by no means exhaustive and one may seek to derive additional valid inequalities by exploiting the structure of binary expansion within the constraints of a particular problem, thus potentially expanding the usefulness of this MILP approach to a wider class of problems.

CHAPTER IV

DISCRETIZATION METHODS FOR POOLING PROBLEM

In this chapter, we study different ways of discretizing the pooling problem and solving the discretized problem as a MILP. The discretized problem is a restriction of the original problem and hence provides feasible solutions and upper bounds on the global optimum of the pooling problem. The motivation for studying discretization methods is based on the fact that MILP solvers are more advanced than global optimization solvers and hence it is more likely that a MILP will be solved faster than a BLP or a MIBLP. Chapter 3 studied different MILP solution techniques for a general MIBLP. Here, we first apply some of the methodology from Chapter 3 to the pooling problem. To do so, we discretize a subset of variables appearing in the problem formulation. Since the pooling problem admits two main alternate formulations ($\mathbb{P}$) and ($\mathbb{PQ}$), different variable choices lead to different MILPs. We address these formulations along with their properties in §4.1. Next, we propose MILP discretizations in §4.2 that are shown to possess a network flow interpretation. Our emphasis is on empirically comparing the performance of the MILP formulations that arise by adopting different discretization methods. We solve the different MILP models in §4.3 and compare the quality of the MILP solution value against the best feasible solution found by a global solver.

We expand on some of the notation used in Chapter 3 for binary representations of integers. For a positive integer, $\mathbb{1}(\cdot)$ is the $\{0,1\}$ vector of binary coding and $\ell(\cdot) := \lfloor \log_2 \cdot \rfloor + 1$ is the length of $\mathbb{1}(\cdot)$. The support of binary coding of a positive integer is $\mathbb{1}^+(\cdot)$. As noted in Observation 3.5, $\ell(\cdot) \in \mathbb{1}^+(\cdot)$. Finally, $\mathbb{1}^+_i(\cdot) := \{t \in \mathbb{1}^+(\cdot) \colon t \geq i\}$, for $i = 1, \ldots, \ell(\cdot)$.

4.1 Variable discretizations

In this section, we discretize a subset of variables in the pooling problem. Towards this end, we first review MILP representations of mixed integer bilinear terms. Consider a single

bilinear term given by

$$\mathcal{T} = \{(\chi, \rho, \omega) \in \Re_+ \times \Re_+ \times \Re_+ \colon \omega = \chi\rho, \chi \in [0, a], \rho \in [0, b]\}. \tag{94}$$

For the sake of simplicity, we have assumed the lower bounds on χ and ρ to be zero.

Now suppose that we discretize ρ, i.e. restrict ρ to take only integer values within its bounds $[0, b]$. This gives us another set $\mathcal{X} \subset \mathcal{T}$ where

$$\mathcal{X} := \{(\chi, \rho, \omega) \in \Re_+ \times \mathbb{Z}_+ \times \Re_+ \colon \omega = \chi\rho, \chi \in [0, a], \rho \in [0, b]\}. \tag{95}$$

Thus, $\mathcal{X}$ is a restriction of $\mathcal{T}$. Substituting $\mathcal{X}$ for every occurrence of $\mathcal{X}$ gives a MIBLP restriction of BLP. Note that for $b > 1$, $\mathcal{X} \subset \mathcal{M}(\mathcal{X}) = \text{conv}(\mathcal{X})$ as shown in Proposition 3.1.

The general integer restriction on ρ, namely $\rho \in \{0, 1, \ldots, b\}$, can be equivalently written as a disjunction $\rho \in \cup_{r=0}^{b}\{r\}$ and hence we have

$$\mathcal{X} = \bigcup_{r=0}^{b} \{(\chi, \rho, \omega) \colon \omega = r\chi, \rho = r, \chi \in [0, a]\}.$$

The extended formulation for this disjunction of polytopes is given by

$$\begin{aligned} \mathcal{U}(\mathcal{X}) := \Big\{(\chi, \rho, \omega, z, \nu) \colon \omega = \sum_{r=0}^{b} r\nu_r, \quad \rho = \sum_{r=0}^{b} r z_r, \\ \sum_{r=0}^{b} z_r = 1, \\ (\chi, z_r, \nu_r) \in \mathcal{M}(\{\nu_r = \chi z_r\}), \quad r = 1, \ldots, b \\ z_r \in \{0, 1\}, r = 1, \ldots, b, \ \chi \in [0, a]\Big\}. \end{aligned} \tag{96}$$

Upon observing that $\rho = \sum_{r=0}^{b} r z_r$ can be interpreted as the base-1 expansion of the general integer variable ρ, we refer to $\mathcal{U}(\mathcal{X})$ as the *unary reformulation* of $\mathcal{X}$ (cf. (54)).

Note that $z_r \in \{0, 1\}, \forall r$, and $\sum_r z_r = 1$ imply a SOS-1 constraint on the z variables, which can be reformulated using a logarithmic number of binary variables and constraints

as shown by Vielma and Nemhauser [108]. Let this log SOS-1 model be denoted by $\mathcal{L}(\mathcal{X})$.

$$\begin{aligned}\mathcal{L}(\mathcal{X}) := \Big\{(\chi,\rho,\omega,z,\nu,\delta)\colon\ & \omega = \sum_{r=0}^{b} r\nu_r, \quad \rho = \sum_{r=0}^{b} r z_r,\\ & \sum_{i=0}^{b} z_r = 1,\\ & (\chi, z_r, \nu_r) \in \mathcal{M}(\{\nu_r = \chi z_r\}), \quad r = 1,\dots,b\\ & \sum_{\substack{r=1:\\ t\in\mathbb{1}^+(r-1)}}^{b} z_r \le \delta_t, \quad t = 1,\dots,\ell(b)\\ & \sum_{\substack{r=1:\\ t\notin\mathbb{1}^+(r-1)}}^{b} z_r \le 1-\delta_t, \quad t = 1,\dots,\ell(b)\\ & z_r \in [0,1], r = 1,\dots,b,\ \delta_t \in \{0,1\}, t = 1,\dots,\ell(b),\ \chi \in [0,a]\Big\}.\end{aligned} \tag{97}$$

Although $\mathcal{L}(\mathcal{X})$ has more variables and constraints than $\mathcal{U}(\mathcal{X})$, it has fewer $\{0,1\}$ variables which can always be an advantage while trying to solve the problem to optimality. We refer to $\mathcal{L}(\mathcal{X})$ as the *log unary reformulation* of $\mathcal{X}$.

Using the base-2 expansion of ρ (cf. (55)) leads to the following *binary reformulation* of $\mathcal{X}$.

$$\begin{aligned}\mathcal{B}(\mathcal{X}) := \Big\{(\chi,\rho,\omega,z,\nu)\colon\ & \omega = \sum_{r=1}^{\ell(b)} 2^{r-1}\nu_r, \quad \rho = \sum_{r=1}^{\ell(b)} 2^{r-1} z_r,\\ & \sum_{r=1}^{\ell(b)} 2^{r-1} z_r \le b,\\ & (\chi, z_r, \nu_r) \in \mathcal{M}(\{\nu_r = \chi z_r\}), \quad r = 1,\dots,\ell(b)\\ & z_r \in \{0,1\}, r = 1,\dots,\ell(b),\ \chi \in [0,a]\Big\},\end{aligned} \tag{98}$$

Note that both $\mathcal{L}(\mathcal{X})$ and $\mathcal{B}(\mathcal{X})$ have the same number of binary variables with the former having more variables and constraints in total.

Substituting any one of $\mathcal{U}(\mathcal{X})$, $\mathcal{L}(\mathcal{X})$, or $\mathcal{B}(\mathcal{X})$ for every occurrence of $\mathcal{X}$ produces a MILP restriction of BLP. We are now ready to use the forgoing ideas in the context of the pooling problem. We consider the p- and pq-formulations and their discretized counterparts.

Consider the pq-formulation formulation ($\mathbb{PQ}$). Each bilinear term is of the form $v_{ilj} = q_{il}y_{lj}$ for some $l \in L, i \in I_l, j \in L \cup J$. Hence, the set corresponding to $\mathcal{T}$ is

$$\mathcal{QT}_{ilj} := \{(q_{il}, y_{lj}, v_{ilj}) \colon v_{ilj} = q_{il}y_{lj},\ q_{il} \in [0,1],\ y_{lj} \in [0, u_{lj}]\}. \tag{99a}$$

for any $l \in L, i \in I_l, j \in L \cup J$. There are two choices for discretization : either $\rho = q_{il}$ or $\rho = y_{lj}$. Similarly, the set representing a single bilinear term in the p-formulation ($\mathbb{P}$) is

$$\mathcal{PT}_{lkj} := \{(p_{lk}, y_{lj}, w_{lkj}) \colon w_{lkj} = p_{lk}y_{lj},\ p_{lk} \in [\underline{p}_{lk}, \bar{p}_{lk}],\ y_{lj} \in [0, u_{lj}]\}, \tag{99b}$$

for any $l \in L, k \in K, j \in L \cup J$ and we may discretize either p_{lk} or y_{lj}.

The notation for representing the various discretized sets in the pooling problem is given in Table 17.

Table 17: Nomenclature for bilinear sets.

Formulation	Indexing	Bilinear term	Discretized variable	Set
($\mathbb{P}$)	$l \in L, k \in K, j \in L \cup J$	$w_{lkj} = p_{lk}y_{lj}$	none y_{lj} (flow) p_{lk} (spec)	$\mathcal{PT}_{lkj}$ $\mathcal{FPX}_{lkj}$ $\mathcal{SPX}_{lkj}$
	none	$w_{lkj} = p_{lk}y_{lj}, \forall l, k, j$	$y_{lj}, \forall l, j$ $p_{lk}, \forall l, k$	$\mathbb{FP}$ $\mathbb{SP}$
($\mathbb{PQ}$)	$l \in L, i \in I_l, j \in L \cup J$	$v_{ilj} = q_{il}y_{lj}$	none y_{lj} (flow) q_{il} (ratio)	$\mathcal{QT}_{ilj}$ $\mathcal{FQX}_{ilj}$ $\mathcal{RQX}_{ilj}$
	none	$v_{ilj} = q_{il}y_{lj}, \forall i, l, j$	$y_{lj}, \forall l, j$ $q_{il}, \forall i, l$	$\mathbb{FPQ}$ $\mathbb{RPQ}$

4.1.1 Flow discretization

The flow discretized model is obtained by discretizing y_{lj} within $[0, u_{lj}]$, for $l \in L, j \in L \cup J$. Thus, for any $l \in L, k \in K, i \in I_l, j \in L \cup J$, we have

$$\mathcal{FPX}_{lkj} := \{(p_{lk}, y_{lj}, w_{lkj}) \in \Re_+ \times \mathbb{Z}_+ \times \Re_+ \colon (p_{lk}, y_{lj}, w_{lkj}) \in \mathcal{PT}_{lkj}\} \tag{100a}$$

$$\mathcal{FQX}_{ilj} := \{(q_{il}, y_{lj}, v_{ilj}) \in \Re_+ \times \mathbb{Z}_+ \times \Re_+ \colon (q_{il}, y_{lj}, v_{ilj}) \in \mathcal{QT}_{ilj}\}. \tag{100b}$$

The flow discretized feasible set for $(\mathbb{P})$ is denoted by $\mathbb{FP}$ and its binary MILP reformulation is $\mathcal{B}(\mathbb{FP})$. For $(\mathbb{PQ})$, it is $\mathbb{FPQ}$ and $\mathcal{B}(\mathbb{FPQ})$, respectively. Since the range $[0, u_{lj}]$ of y_{lj} is typically of high order, we only consider the binary expansion of y_{lj} in order to avoid adding too many extra $\{0,1\}$ variables. We assume that pool capacity C_l and arc capacity u_{lj} are integers, for all $l \in L, j \in L \cup J$, otherwise they can be replaced with $\lfloor C_l \rfloor$ and $\lfloor u_{lj} \rfloor$, respectively. Observe that we do not have to discretize all the flow variables in the original formulation. Only flows on outgoing arcs from each pool are discretized, since they give rise to bilinear terms in the sets $\mathcal{FPX}_{lkj}$ and $\mathcal{FQX}_{ilj}$.

We now discuss some valid inequalities for the flow discretized models. Consider $\mathcal{B}(\mathbb{FPQ})$ and choose a pool $l \in L$. Let j be some outgoing arc from this pool l. The flow variable y_{lj} is subjected to binary expansion as $y_{lj} = \sum_{r=1}^{\ell(u_{lj})} 2^{r-1} z_{rlj} \leq u_{lj}$. Since $q_l \in \Delta^{|I_l|}$, Corollary 3.2 and Proposition 3.9 then imply that the convex hull of $\cap_{i \in I_l} \mathcal{B}(\mathcal{FQX}_{ilj})$ is defined by the following nontrivial inequalities

$$\sum_{r' \in \mathbb{1}_r^+(u_{lj})} \nu_{r'ilj} \leq \left(|\, \mathbb{1}_r^+(u_{lj})| - 1\right) q_{il}, \qquad r \notin \mathbb{1}^+(u_{lj}), i \in I_l \tag{101a}$$

$$\sum_{i \in I_l} \nu_{rilj} = z_{rlj}, \qquad r = 1, \ldots, \ell(u_{lj}). \tag{101b}$$

Note that $\cap_{i \in I_l} \mathcal{B}(\mathcal{FQX}_{ilj})$ considers only one outgoing arc j. We may include all the outgoing arcs from pool l. In this case, the inequalities (101), for all $j \in L \cup J$, suffice to describe the convex hull of $\cap_{j \in L \cup J} \cap_{i \in I_l} \mathcal{B}(\mathcal{FQX}_{ilj})$, due to Proposition 3.4.

For $\mathcal{B}(\mathbb{FP})$, the inequalities are

$$\sum_{r' \in \mathbb{1}_r^+(u_{lj})} \nu_{r'lkj} - \underline{p}_{lk} \sum_{r' \in \mathbb{1}_r^+(u_{lj})} z_{r'lj} \leq \left(|\, \mathbb{1}_r^+(u_{lj})| - 1\right)(p_{lk} - \underline{p}_{lk}), \qquad r \notin \mathbb{1}^+(u_{lj}) \tag{102a}$$

$$\bar{p}_{lk} \sum_{r' \in \mathbb{1}_r^+(u_{lj})} z_{r'lj} - \sum_{r' \in \mathbb{1}_r^+(u_{lj})} \nu_{r'lkj} \leq \left(|\, \mathbb{1}_r^+(u_{lj})| - 1\right)(\bar{p}_{lk} - p_{lk}), \qquad r \notin \mathbb{1}^+(u_{lj}) \tag{102b}$$

obtained by multiplying the minimal covers of u_{lj} with $p_{lk} - \underline{p}_{lk}$ and $\bar{p}_{lk} - p_{lk}$. By Proposition 3.4, (102) for all $j \in L \cup J$ are the nontrivial facets for the convex hull of $\cap_j \mathcal{B}(\mathcal{FPX}_{lkj})$.

Although the inequalities (101) and (102) are valid to $\mathcal{B}(\mathbb{FPQ})$ and $\mathcal{B}(\mathbb{FP})$, respectively, they are obtained using minimal covers for each individual outgoing arc. The constraint $\sum_{j \in L \cup J} \sum_r 2^{r-1} z_{rlj} \leq C_l$, that arises from the pool capacity constraints (2b) and binary expansion of y_{lj}, for all $j \in L \cup J$, was relaxed while deriving (101) and (102). We now derive new valid inequalities from this capacity constraint.

4.1.1.1 Binary expansion of GUB constraints

Consider a set $\mathcal{K} := \{\rho \in \mathbb{Z}^n_+ \colon \sum_{j=1}^n \rho_j \leq b\}$ defined by a generalized upper bound (GUB) constraint. Assume w.l.o.g. that b is integral. The set $\mathcal{K}$ can be construed as the discretized counterpart of the pool capacity constraint (2b) with ρ_j corresponding to the arc flow y_{lj} and $b = \lfloor C_l \rfloor$. Clearly, $\text{conv}(\mathcal{K}) = \text{relax}(\mathcal{K})$. We wish to obtain valid inequalities for the binary reformulation of $\mathcal{K}$,

$$\mathcal{B}(\mathcal{K}) = \left\{ z \in \{0,1\}^{n\ell(b)} \colon \sum_{j=1}^{n} \sum_{t=1}^{\ell(b)} 2^{t-1} z_{jt} \leq b \right\}. \tag{103}$$

Note that the defining knapsack can be rearranged to give

$$\mathcal{B}(\mathcal{K}) = \left\{ z \in \{0,1\}^{n\ell(b)} \colon \sum_{t=1}^{\ell(b)} 2^{t-1} \sum_{j=1}^{n} z_{jt} \leq b \right\}.$$

In the binary reformulation set $\mathcal{B}(\mathcal{FQX}_{ilj})$, the number of extra $\{0,1\}$ variables is equal to $\ell(u_{lj})$. For simplicity, we have made the following assumption about the upper bounds in $\mathcal{K}$.

Assumption 4.1. Let u_j be an upper bound on ρ_j, for all j. We assume all the upper bounds to be of the same order in the sense that $\ell(u_j) = \ell(b)$, for all $j = 1, \ldots, n$.

This assumption allows using $\ell(b)$ many $\{0,1\}$ variables for each j. Although this assumption is not w.l.o.g., we may use $\ell(u_j)$ many $\{0,1\}$ variables for j and our forthcoming results can be generalized by appropriately defining an equivalence class for each index t, i.e. $[t] := \{j \in \{1, \ldots, n\} \colon t \leq \ell(u_j)\}$ is those subset of arcs j whose upper bound u_j is

large enough such that the binary coding of u_j contains t. However, for ease of exposition, we have the previous assumption.

We first provide a disjunctive characterization for $\mathcal{B}(\mathcal{K})$. For any $t \in \mathbb{1}^+(b)$, define

$$\begin{aligned}
\mathcal{B}^t(\mathcal{K}) &:= \left\{ z \in \mathcal{B}(\mathcal{K}) \colon \sum_{j=1}^{n} z_{jr} = 1, \quad \forall r \in \mathbb{1}^+_t(b) \setminus t \right\}, \\
\mathcal{B}^{t,1}(\mathcal{K}) &:= \left\{ z \in \mathcal{B}^t(\mathcal{K}) \colon \sum_{j=1}^{n} z_{jt} = 1 \right\}, \qquad \mathcal{B}^{t,0}(\mathcal{K}) := \left\{ z \in \mathcal{B}^t(\mathcal{K}) \colon \sum_{j=1}^{n} z_{jt} = 0 \right\}.
\end{aligned}$$

Observe that $\mathcal{B}^{\ell(b)}(\mathcal{K}) = \mathcal{B}(\mathcal{K})$.

Proposition 4.1. *For any $t \in \mathbb{1}^+(b)$, the inequality $\sum_{j=1}^{n} z_{jt} \leq 1$ is valid to $\mathcal{B}^t(\mathcal{K})$. Moreover,*

$$\mathcal{B}(\mathcal{K}) = \mathcal{B}^{i_1,1}(\mathcal{K}) \cup \left(\bigcup_{t \in \mathbb{1}^+(b)} \mathcal{B}^{t,0}(\mathcal{K}) \right),$$

where $i_1 = \min\{t' \colon t' \in \mathbb{1}^+(b)\}$.

Proof. The proof of the first statement is by induction on t. We know that $\ell(b) \in \mathbb{1}^+(b)$. The defining inequality of $\mathcal{B}(\mathcal{K})$ can be rewritten as

$$\sum_{r=1}^{\ell(b)-1} 2^{r-1} \sum_{j=1}^{n} z_{jr} \; + \; 2^{\ell(b)-1} \sum_{j=1}^{n} z_{j\ell(b)} \; \leq \; b.$$

Divide both sides by $2^{\ell(b)-1}$ and observe that $2^{\ell(b)-1} \leq b \leq 2^{\ell(b)}$. Applying the Chvatal-Gomory rounding procedure [cf. 79, §II.1.1] yields the desired inequality $\sum_{j=1}^{n} z_{j\ell(b)} \leq 1$.

Now set $\sum_{j=1}^{n} z_{j\ell(b)} = 1$ and consider $\mathcal{B}^{\ell(b)-1}(\mathcal{K})$. The defining inequality for this set is

$$\sum_{r=1}^{\ell(b)-1} 2^{r-1} \sum_{j=1}^{n} z_{jr} \; \leq \; b - 2^{\ell(b)-1}.$$

Let $k := \max\{t' \in \mathbb{1}^+(b) \colon t' < \ell(b)\}$. Note that $\sum_{t' \in \mathbb{1}^+(b) \colon t' \leq k} 2^{t'-1} = b - 2^{\ell(b)-1}$. Hence, $k = \ell(b - 2^{\ell(b)-1})$. Then, for any $k < r < \ell(b)$, $z_{jr} = 0$, for all $j = 1, \ldots, n$. Continuing this argument iteratively, we get the following claim : for any $t \in \mathbb{1}^+(b)$ and $k := \max\{t' \in \mathbb{1}^+(b) \colon t' < t\}$, we must have $z \in \mathcal{B}^{t,1}(\mathcal{K})$ implies $\sum_{j=1}^{n} z_{jr} = 0$, for all $r > k, r \notin \mathbb{1}^+(b)$. The defining inequality for $\mathcal{B}^t(\mathcal{K})$ is then given by

$$\sum_{r=1}^{t-1} 2^{r-1} \sum_{j=1}^{n} z_{jr} \; + \; 2^{t-1} \sum_{j=1}^{n} z_{jt} \; \leq \; b - \sum_{t' \in \mathbb{1}^+_t(b) \setminus t} 2^{t'-1}.$$

Dividing by 2^{t-1} and applying CG rounding implies validity of $\sum_{j=1}^{n} z_{jt} \leq 1$ to $\mathcal{B}^t(\mathcal{K})$.

From the validity of $\sum_{j=1}^{n} z_{jt} \leq 1$, it follows that $\mathcal{B}^t(\mathcal{K}) = \mathcal{B}^{t,0}(\mathcal{K}) \cup \mathcal{B}^{t,1}(\mathcal{K})$. The disjunction is then straightforward. □

Before presenting valid inequalities for $\text{conv}(\mathcal{B}(\mathcal{K}))$, consider the following notation that takes into account the multiplicities of each $i \in \{1, \ldots, \ell(b)\}$ across $j \in \{1, \ldots, n\}$.

Definition 4.1. For any $i = 1, \ldots, \ell(b)$, let $\Omega_i := \{1, \ldots, n\}^{|\, \mathbb{1}_i^+(b)|}$ be a bounded integer lattice of dimension $|\, \mathbb{1}_i^+(b)|$. Each point in this lattice is a vector of indices, denoted as $(j_i, \{j_t\} \colon \forall t \in \mathbb{1}_i^+(b) \setminus \ell(b))$ for some suitable indices $j_i, \{j_t\}_t \in \{1, \ldots, n\}$.

We illustrate the set Ω_i with a simple example.

Example 4.1. Let $n = 2$ and $b = 14 = 2 + 4 + 8$ such that $\ell(b) = 4$. Then,

$$\mathbb{1}_1^+(14) = \{2,3,4\}, \mathbb{1}_2^+(14) = \{2,3,4\}, \mathbb{1}_3^+(14) = \{3,4\}, \mathbb{1}_4^+(14) = \{4\}, \qquad \text{and}$$

$$\begin{aligned}
\Omega_1 &= \{(1,1,1),(2,1,1),(1,2,1),(2,2,1),(1,1,2),(2,1,2),(1,2,2),(2,2,2)\} \\
\Omega_2 &= \{(1,1,1),(2,1,1),(1,2,1),(2,2,1),(1,1,2),(2,1,2),(1,2,2),(2,2,2)\} \\
\Omega_3 &= \{(1,1),(1,2),(2,1),(2,2)\} \\
\Omega_4 &= \{1,2\}.
\end{aligned}$$

The set $\mathcal{B}(\mathcal{K})$ is symmetric upto permutations of indices with the same coefficient. Hence, given a index $i \in \{1, \ldots, \ell(b)\}$ and a minimal cover inequality (cf. §3.3.2) for binary expansion of b, every element of Ω_i produces a valid inequality to $\text{conv}(\mathcal{B}(\mathcal{K}))$.

Proposition 4.2. *For any $i = 1, \ldots, \ell(b)$ and $(j_i, \{j_t\}_t) \in \Omega_i$, the extended minimal cover*

$$z_{j_i i} \; + \sum_{t \in \mathbb{1}_i^+(b) \setminus \ell(b)} z_{j_t t} \; + \sum_{j'=1}^{n} z_{j' \ell(b)} \; \leq \; |\, \mathbb{1}_i^+(b)| \tag{104}$$

is valid to $\text{conv}(\mathcal{B}(\mathcal{K}))$.

Proof. For $i \notin \mathbb{1}^+(b)$, we have already shown in Proposition 3.7 that $(j_i, i) \cup \{(j_t, t) \colon t \in \mathbb{1}_i^+(b)\}$ is a minimal cover. Since $z_{j'\ell(b)}$ has the largest coefficient for any j' and $\ell(b) \in \mathbb{1}^+(b)$, the proposed inequality is a extended minimal cover. For $i \in \mathbb{1}^+(b)$, a similar argument as in Proposition 3.7 proves that $(j_i, i) \cup (j_i', i) \cup \{(j_t, t) \colon t \in \mathbb{1}_i^+(b), t > i\}$ is indeed a minimal cover. Validity is then due to Nemhauser and Wolsey [79], Proposition II.2.2.2. □

Next, we present families of inequalities that can be proven to be valid via the split disjunction $\{\sum_{j=1}^{n} z_{j\ell(b)} = 0\} \vee \{\sum_{j=1}^{n} z_{j\ell(b)} = 1\}$ for $\mathcal{B}(\mathcal{K})$.

Proposition 4.3. *The following two classes of split inequalities are valid to* $\text{conv}(\mathcal{B}(\mathcal{K}))$.

1. *For any* $i = 1, \ldots, \ell(b)$, $(j_i, \{j_t\}_t) \in \Omega_i$, *and* $r \notin \mathbb{1}^+(b)$ *such that* $r > \max\{t\colon t \in \mathbb{1}^+(b) \setminus \ell(b)\}$,

$$z_{j_i i} + \sum_{t \in \mathbb{1}_i^+(b) \setminus \ell(b)} z_{j_t t} + \sum_{j'=1}^{n} z_{j' r} + \left\lfloor \frac{b}{2^{r-1}} \right\rfloor \sum_{j'=1}^{n} z_{j' \ell(b)} \;\leq\; |\,\mathbb{1}_i^+(b)| - 1 + \left\lfloor \frac{b}{2^{r-1}} \right\rfloor \tag{105a}$$

2. *For any* $i \notin \mathbb{1}^+(b)$, $(j_i, \{j_t\}_t) \in \Omega_i$, *and* $\tilde{j} \in \{1, \ldots, n\}, \tilde{t} = \max\{t\colon t \in \mathbb{1}^+(b) \setminus \ell(b)\}$,

$$z_{j_i i} \;+ \sum_{t \in \mathbb{1}_i^+(b) \setminus \ell(b)} z_{j_t t} \;+\; z_{\tilde{j}\tilde{t}} \;+\; 2 \sum_{j'=1}^{n} z_{j' \ell(b)} \;\leq\; |\,\mathbb{1}_i^+(b)| + 1. \tag{105b}$$

Proof. For (105a), setting $\sum_{j=1}^{n} z_{j\ell(b)} = 1$ implies $z_{j'r} = 0, \forall j'$, as discussed in Proposition 4.1, and hence we get

$$z_{j_i i} \;+ \sum_{t \in \mathbb{1}_i^+(b) \setminus \ell(b)} z_{j_t t} \;\leq\; |\,\mathbb{1}_i^+(b)| - 1,$$

which is valid due to Proposition 4.2. Now let $\sum_{j=1}^{n} z_{j\ell(b)} = 0$. Note that $\sum_{j'=1}^{n} z_{j'r} \leq \lfloor b/2^{r-1} \rfloor$ is a valid bound for $\mathcal{B}(\mathcal{K})$. If $z_{j_i i} + \sum_{t \in \mathbb{1}_i^+(b) \setminus \ell(b)} z_{j_t t} = |\,\mathbb{1}_i^+(b)| - 1$, then validity follows. Suppose that $z_{j_i i} + \sum_{t \in \mathbb{1}_i^+(b) \setminus \ell(b)} z_{j_t t} = |\,\mathbb{1}_i^+(b)|$. Assume, for sake of contradiction, that $\sum_{j'=1}^{n} z_{j'r} = \lfloor b/2^{r-1} \rfloor$. Note that by construction, $\lfloor b/2^{r-1} \rfloor = 2^{\ell(b)-r}$. Since $z \in \mathcal{B}(\mathcal{K})$, we have

$$\begin{aligned}
\sum_j \sum_t 2^{t-1} z_{jt} \;&\geq\; 2^{r-1} \left\lfloor \frac{b}{2^{r-1}} \right\rfloor \;+\; 2^{i-1} \;+ \sum_{t \in \mathbb{1}_i^+(b) \setminus \ell(b)} 2^{t-1} \\
&=\; 2^{r-1} \frac{2^{\ell(b)}}{2^r} \;+\; 2^{i-1} \;+ \sum_{t \in \mathbb{1}_i^+(b) \setminus \ell(b)} 2^{t-1} \\
&=\; 2^{\ell(b)-1} \;+\; 2^{i-1} \;+ \sum_{t \in \mathbb{1}_i^+(b) \setminus \ell(b)} 2^{t-1} \\
&=\; 2^{i-1} \;+ \sum_{t \in \mathbb{1}_i^+(b)} 2^{t-1} \\
&>\; b,
\end{aligned}$$

where the first equality is due to $\lfloor b/2^{r-1} \rfloor = 2^{\ell(b)-r}$ and the last strict inequality is due to the fact that $(j_i, i) \cup \{(j_t, t)\colon t \in \mathbb{1}_i^+(b)\}$ is a cover. Thus, we have reached a contradiction

to the feasibility of z. Hence, it must be that $\sum_{j'=1}^{n} z_{j'r} \leq \lfloor b/2^{r-1} \rfloor - 1$. This completes the validity proof for (105a).

Now consider (105b). If $\sum_{j=1}^{n} z_{j\ell(b)} = 1$, then $z_{j_{\tilde{t}}\tilde{t}} + z_{\tilde{j}\tilde{t}} \leq 1$ is valid to $\mathcal{B}^{\tilde{t}}(\mathcal{K})$ by Proposition 4.1, and hence the inequality is valid by Proposition 4.2. Else $\sum_{j=1}^{n} z_{j\ell(b)} = 0$, and then (105b) is trivially true because $z_{jt} \leq 1$, for all j, t. □

The inequalities in (104) and (105) can be separated greedily for each $i \in \{1, \ldots, \ell(b)\}$. In particular, we can choose $(j_i, \{j_t\}_t) \in \Omega_i$ such that at any incumbent point z, $j_i = \arg\max\{z_{ji} \colon j = 1, \ldots, n\}$ and $j_t = \arg\max\{z_{jt} \colon j = 1, \ldots, n\}$, for $t \in \mathbb{1}_i^+(b) \setminus \ell(b)$. Multiplying (104) and (105) by q_{il} on both sides, as done for minimal covers of u_{lj} in (101), produces cutting planes for $\mathcal{B}(\mathbb{FPQ})$. Similarly, multiplying by p_{lk} gives cutting planes for $\mathcal{B}(\mathbb{FP})$.

4.1.2 Ratio and specification discretization

Here we discretize the non-flow variables in the pooling problem. For $(\mathbb{PQ})$, this entails discretizing ratio variables q_{il} for all $l \in L, i \in I_l$. Although each ratio q_{il} can be discretized into different intervals, for the ease of exposition, we assume that all the ratios are uniformly discretized into $n \geq 1$ intervals of equal length within $[0, 1]$. Note that unlike §4.1.1 where discretizing flows to integer values within their respective bounds seemed like a reasonable method, in this case there is no clear intuition behind a suitable choice of n. In our computations, we experiment with different values of n. The discretized single bilinear set is

$$\mathcal{RQX}_{ilj} := \{(q_{il}, y_{lj}, v_{ilj}) \colon v_{ilj} = q_{il} y_{lj},\ nq_{il} \in [0, n] \cap \mathbb{Z},\ y_{lj} \in [0, u_{lj}]\}. \tag{106}$$

Including $\mathcal{RQX}_{ilj}$, for all $l \in L, i \in I_l, j \in L \cup J$, gives us the ratio discretized feasible set for the pq-formulation, denoted by $\mathbb{RPQ}$. Applying the MILP reformulations to $\mathcal{RQX}_{ilj}$ gives us $\mathcal{U}(\mathcal{RQX}_{ilj})$, $\mathcal{L}(\mathcal{RQX}_{ilj})$, and $\mathcal{B}(\mathcal{RQX}_{ilj})$, respectively. Consequently, the MILP models for $\mathbb{RPQ}$ are $\mathcal{U}(\mathbb{RPQ})$, $\mathcal{L}(\mathbb{RPQ})$, and $\mathcal{B}(\mathbb{RPQ})$, respectively.

For $(\mathbb{P})$, the specification discretized model is obtained by discretizing p_{lk} within its

bounds $[\underline{p}_{lk}, \bar{p}_{lk}]$ in the set $\mathcal{PT}_{lkj}$.

$$\mathcal{SPX}_{lkj} := \big\{(p_{lk}, y_{lj}, w_{lkj})\colon w_{lkj} = p_{lk}y_{lj},\ y_{lj} \in [0, u_{lj}], \\ np_{lk} \in \bigcup_{r=0}^{n} \{n\underline{p}_{lk} + (\bar{p}_{lk} - \underline{p}_{lk})r\} \big\}. \tag{107}$$

Similar to ratio discretization, we experiment with different values of n. The spec discretized feasible set for the p-formulation is denoted by $\mathbb{SP}$ and the MILP models are $\mathcal{U}(\mathbb{SP}), \mathcal{L}(\mathbb{SP})$, and $\mathcal{B}(\mathbb{SP})$, respectively. The unary model for spec discretization was first studied by Pham et al. [82].

For the binary MILPs $\mathcal{B}(\mathbb{SP})$ and $\mathcal{B}(\mathbb{RPQ})$, we can derive cutting planes similar to the ones presented in (101) and §4.1.1.1. However, note that (101) defined the convex hull of $\cap_j \cap_i \mathcal{FQX}_{ilj}$ because the nondiscretized variable q_l belonged to a simplex $\Delta^{|I_l|}$. For spec and ratio discretization, the nondiscretized variable is $y_{l\cdot} \in \mathcal{F}_l = \{y_l\colon \sum_{j \in L \cup J} y_{lj} \leq C_l, y_{lj} \in [0, u_{lj}]\}$, which is not a simplex. Hence, analogues of (101) may not give us the convex hull of $\cap_j \cap_i \mathcal{RQX}_{ilj}$ and $\cap_j \cap_k \mathcal{SPX}_{lkj}$.

4.2 Network flow MILPs

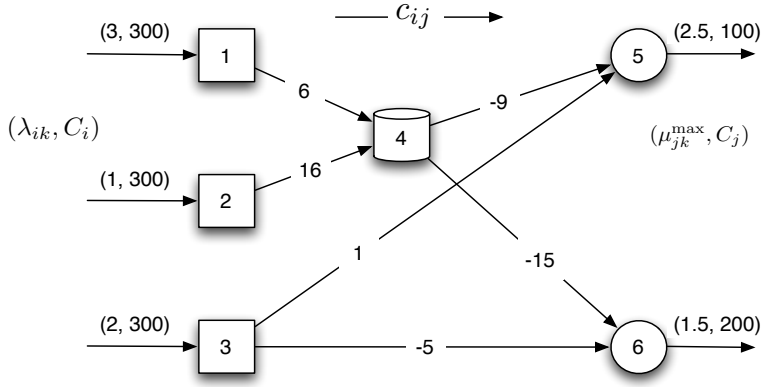
In this section, we discuss two MILP formulations that have a network flow interpretation. Both these models are defined on an expanded network. The network for the second model has exponentially many new nodes. The first discretization arises while eliminating bilinear terms at each pool whereas the second model is a reformulation of ratio discretization $\mathbb{RPQ}$.

4.2.1 Discretizing consistency requirements at each pool

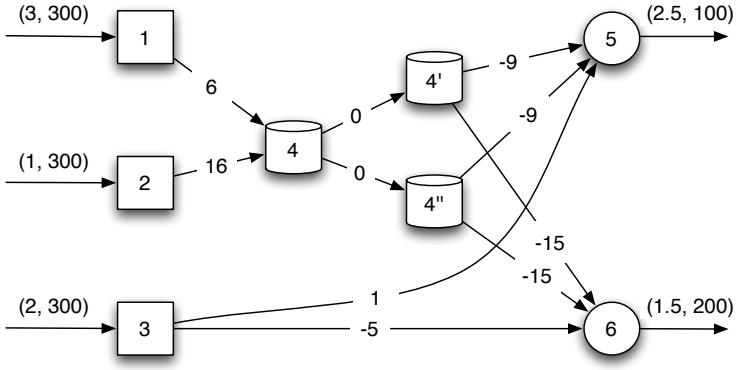
Consider the p-formulation formulation ($\mathbb{P}$) for the pooling problem. Bilinear terms arise at each pool and are of the form $w_{lkj} = p_{lk}y_{lj}$, i.e. a product between the specification $k \in K$ at pool $l \in L$ and the outflow on arc $(l, j) \in \mathcal{A}$. The set $\mathcal{PT}_{lkj}$ in (99b) defines this bilinear term. The physical interpretation of this bilinear term is that w_{lkj} is the absolute amount of specification k flowing on arc (l, j). Note that the specification variables p_{lk}'s have no costs associated with them and do not appear in the objective function. Hence, the sole purpose of introducing the p_{lk} variables in the p-formulation is to ensure consistency among all the outgoing arcs from this pool l. In particular, p_{lk} is independent of j and we want

that every outgoing arc (l, j) carry the same concentration p_{lk} of this specification k. Thus, we are enforcing the ratio w_{lkj}/y_{lj} to be equal to p_{lk}, for all $j \in L \cup J$.

Now we propose a discretization of $(\mathbb{P})$ that enforces the ratios w_{lkj}/y_{lj} to be equal for all j, without explicitly introducing p_{lk}. Consider the pooling problem on a directed graph $G = (\mathcal{N}, \mathcal{A})$. Let n be a chosen level of discretization. We will define the proposed MILP restriction on a expanded directed graph $\tilde{G} = (\tilde{\mathcal{N}}, \tilde{\mathcal{A}})$. For each pool $l \in L$, we create n duplicate nodes and let $\tilde{L}_l$ be this set of duplicate nodes. Thus, $\tilde{L} = \cup_{l \in L} \tilde{L}_l$ is the set of duplicate pools and $\tilde{\mathcal{N}} = \mathcal{N} \cup \tilde{L}$. For every duplicate pool $\tau \in \tilde{L}_l$, introduce an arc (l, τ) and $|L \cup J|$ many arcs of the form $(\tau, j), \forall j \in L \cup J$. Delete the old arcs $(l, j), \forall j \in L \cup J$. This expanded graph $\tilde{G}$ is illustrated in Figure 9.



(a) The original `Haverly1` instance [60].



(b) Uniform discretized model $\mathbb{EP}$ with $n = 2$.

Figure 9: Enforcing consistency requirements at each pool using a expanded network.

We retain the flow variables y_{ij} on the original arc set $\mathcal{A}$ and also introduce new variables $y_{i'j'}$ on the expanded arc set $\tilde{\mathcal{A}}$. Suppose that we restrict the outflows from pool l to be

distributed in a manner such that for any $\tau \in \tilde{L}_l$, we have

$$\begin{aligned} \gamma_{l\tau} \sum_{i \in I \cup L} y_{il} &= y_{l\tau} \\ &= \sum_{j \in L \cup J} y_{\tau j} \end{aligned} \tag{108a}$$

where the second equality is due to flow balance at τ and $\gamma_{l\tau} \in (0, 1]$ such that $\sum_{\tau \in \tilde{L}_l} \gamma_{l\tau} = 1$. Thus, the τ^{th} duplicate pool receives $\gamma_{l\tau}^{th}$ fraction of the total incoming flow at pool l. Similarly, we enforce the absolute amounts of any specification k to be distributed as

$$\gamma_{l\tau} \left[\sum_{i \in I} \lambda_{ik} y_{il} + \sum_{l' \in L} w_{l'kl} \right] = \sum_{j \in L \cup J} w_{\tau kj} \qquad k \in K. \tag{108b}$$

To maintain correctness between the variables in G and $\tilde{G}$, we also need that for every $j \in L \cup J$

$$\sum_{\tau \in \tilde{L}_l} y_{\tau j} = y_{lj} \tag{109a}$$

$$\sum_{\tau \in \tilde{L}_l} w_{\tau kj} = w_{lkj} \qquad k \in K. \tag{109b}$$

Arbitrary splitting of inflows at a pool does not yet guarantee that the same specification p_{lk} is available across all output arcs j. However, if we enforce that each duplicate pool $\tau \in \tilde{L}_l$ be allowed to send flow to only one outgoing arc j, i.e. the outflows from each duplicate pool be SOS-1 constrained, modeled as

$$y_{\tau j} \leq u_{lj} \zeta_{\tau j} \quad \forall j \tag{110a}$$

$$\underline{p}_{lk} y_{\tau j} \leq w_{\tau kj} \leq \bar{p}_{lk} y_{\tau j} \quad \forall j, k \tag{110b}$$

$$\sum_{j \in L \cup J} \zeta_{\tau j} = 1, \quad \zeta_{\tau j} \in \{0, 1\}, \forall j \tag{110c}$$

then we claim that the same specification p_{lk} is available across all output arcs j. We argue this next. Consider a pool $l \in L$ and an outgoing arc (l, j) for some $j \in L \cup J$. Suppose that in $\tilde{G}$, the node j receives positive inflow from $M_j \geq 0$ many duplicate pools in $\tilde{L}_l$. For convenience, let these duplicate pools be indexed by $\{1, \ldots, M_j\}$. Since the outflows from each duplicate pool are SOS-1 and $|\tilde{L}_l| = n$, we have $\sum_{j \in L \cup J} M_j = n$. The available

specification at node j due to pool l is given by the ratio

$$\begin{aligned}\frac{\sum_{\tau=1}^{M_j} w_{\tau kj}}{\sum_{\tau=1}^{M_j} y_{\tau j}} &= \frac{\sum_{\tau=1}^{M_j} \gamma_{l\tau}\left[\sum_{i\in I}\lambda_{ik}y_{il} + \sum_{l'\in L} w_{l'kl}\right]}{\sum_{\tau=1}^{M_j}\gamma_{l\tau}\sum_{i\in I\cup L} y_{il}} \\ &= \frac{\left[\sum_{i\in I}\lambda_{ik}y_{il} + \sum_{l'\in L} w_{l'kl}\right]}{\sum_{i\in I\cup L} y_{il}} \\ &= p_{lk}\end{aligned}$$

where the first equality is using SOS-1 outflows from τ in (108a) and (108b).

Thus, we have the following MILP restriction of ($\mathbb{P}$), denoted as ($\mathbb{EP}$).

$$\begin{aligned}\min_{y,w,\zeta} \quad & \sum_{(i,j)\in\mathcal{A}} c_{ij}y_{ij} \\ \text{s.t.} \quad & y\in\mathcal{F}, \quad (108)-(110) \\ & \sum_{i\in I}\lambda_{ik}y_{il} + \sum_{l'\in L} w_{l'kl} = \sum_{j\in L\cup J} w_{lkj}, \quad l\in L, k\in K \\ & \sum_{i\in I}\lambda_{ik}y_{ij} + \sum_{l\in L} w_{lkj} \le \mu_{jk}^{\max}\sum_{i\in I\cup L} y_{ij}, \quad j\in J, k\in K \\ & \sum_{i\in I}\lambda_{ik}y_{ij} + \sum_{l\in L} w_{lkj} \ge \mu_{jk}^{\min}\sum_{i\in I\cup L} y_{ij}, \quad j\in J, k\in K.\end{aligned} \tag{$\mathbb{EP}$}$$

In the preceding discussion, we have allowed the choice for $\gamma_{l\tau} \in (0,1]$ to be arbitrary upto the requirement that $\sum_{\tau\in\tilde{L}_l}\gamma_{l\tau} = 1$. We now propose two choices for $\gamma_{l\tau}$. Let n be the chosen level of discretization.

Uniform model : Set $\gamma_{l\tau} = 1/n$ for all $\tau\in\tilde{L}_l, l\in L$. Since $|\tilde{L}_l| = n$, this is a valid choice.

We refer to the corresponding discretization of ($\mathbb{P}$) by $\mathbb{UEP}$.

Asymmetric model : Let $\tilde{L}_l$ be an ordered set and $\text{ord}(\tau)$ be the position of an element τ in the set $\tilde{L}_l$. Then, for every $l\in L$, we choose

$$\gamma_{l\tau} = \begin{cases} \frac{1}{2^{\text{ord}(\tau)}} & \text{ord}(\tau) \le n-1 \\ \\ \frac{1}{2^{n-1}} & \text{ord}(\tau) = n. \end{cases}$$

It is easy to verify that $\sum_\tau \gamma_{l\tau} = 1$. The MILP model is denoted by $\mathbb{AEP}$.

Note that $\mathbb{UEP}$ and $\mathbb{AEP}$ are equivalent for $n = 1, 2$. For $n = 1$, there is only one duplicate pool for each original pool and it follows that the outflows from the original pool are constrained to be SOS-1.

We can adopt a similar discretization approach for ($\mathbb{PQ}$). Here, we need to maintain consistency with respect to the incoming flow ratios q_{il} across all outgoing arcs (l, j) from a pool l. The MILP model is analogous with w_{lkj} and $w_{\tau kj}$ replaced by v_{ilj} and $v_{\tau ij}$, respectively. The uniform and asymmetric discretizations are denoted by $\mathbb{UEPQ}$ and $\mathbb{AEPQ}$, respectively.

4.2.2 Exponentially large formulation for ratio discretization

We now discuss another MILP reformulation of $\mathbb{RPQ}$ proposed by Alfaki and Haugland [8]. In this formulation we do not include additional variables v_{ilj} for the bilinear terms $q_{il}y_{lj}$ nor do we add the variables $(z_{ril}, \nu_{rilj}), \forall i, r, l, j$, of §4.1. Instead, for each pool $l \in L$, we explicitly enumerate all the feasible points of the discretized simplex

$$\tilde{\Delta}^{|I_l|} = \left\{ q_l \in \Re_+^{|I_l|} \colon \sum_{i \in I_l} q_{il} = 1,\ nq_{il} \in [0, n] \cap \mathbb{Z}, i \in I_l \right\}, \tag{111}$$

where n is the level of discretization, and create duplicate nodes, one for each feasible point in $\tilde{\Delta}^{|I_l|}$. These duplicate pools inherit properties, such as arc connectivity and node capacity, of its parent pool node. For each duplicate pool an additional binary variable is created which is equal to 1 if and only if the discretized ratio corresponding to that duplicate pool is selected. For each original pool, the binary variables corresponding to its duplicate pools are SOS-1 constrained. Note that the cardinality of $\tilde{\Delta}^{|I_l|}$ is exponentially large and hence this model has exponentially many new $\{0, 1\}$ variables and constraints.

Consider the pooling problem on a directed graph $G = (\mathcal{N}, \mathcal{A})$. We will define the proposed MILP approximation for the pooling problem on a expanded directed graph $\tilde{G} = (\tilde{\mathcal{N}}, \tilde{\mathcal{A}})$. For each pool $l \in L$, let $\tilde{L}_l$ be a set of duplicate pools of cardinality $|\tilde{\Delta}^{|I_l|}|$, i.e one duplicate pool for each point in the discretized simplex $\tilde{\Delta}^{|I_l|}$. Thus, $\tilde{L} = \cup_{l \in L} \tilde{L}_l$ is the set of duplicate pools. Delete the original pool l and set $\tilde{\mathcal{N}} = (\mathcal{N} \setminus L) \cup \tilde{L}$. Denote $\tilde{\Delta}^{|I_l|} = \{\tilde{q}_l^1, \ldots, \tilde{q}_l^{|\tilde{\Delta}^{|I_l|}|}\}$ where each $\tilde{q}_l^\tau$ is a feasible point in $\tilde{\Delta}^{|I_l|}$ that can be enumerated a priori. For every duplicate pool $\tau \in \tilde{L}_l$, introduce $|I \cup L|$ many arcs of the form (i, τ), for all $i \in I \cup L$, and $|L \cup J|$ many arcs of the form (τ, j), for all $j \in L \cup J$. Figure 10 illustrates this model on a simple pooling instance.

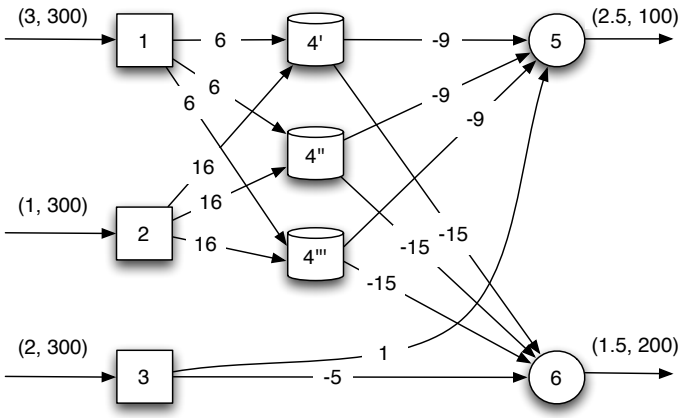


Figure 10: Expanded network MILP $\mathcal{E}(\mathbb{RPQ})$ with $n = 2$ for `Haverly1` instance. $\mathcal{E}(\mathbb{RPQ})$ is a reformulation of $\mathbb{RPQ}$.

Define y_{ij} to be the flow on arc $(i, j) \in \tilde{\mathcal{A}}$. For $l \in L, \tau \in \tilde{L}_l$, let $\zeta_\tau \in \{0, 1\}$ be a binary variable such that $\zeta_\tau = 1$ if and only if $q_{il} = \tilde{q}^\tau_{il}$, for all $i \in I_l$, i.e. the τ^{th} discretized ratio is chosen at pool l. Note that we do not need to add q variables in this MILP. There are two types of combinatorial constraints: the first one an SOS-1 constraint to ensure exactly one discretization is chosen for pool l, and the second one a variable upper bound constraint such that $y_{\tau j} = 0$ if $\zeta_{\tau l} = 0$.

$$\sum_{\tau \in \tilde{L}_l} \zeta_{\tau l} = 1, \quad l \in L, \qquad \zeta_{\tau l} \in \{0, 1\},\ l \in L, \tau \in \tilde{L}_l \tag{112a}$$

$$y_{\tau j} \leq C_l \zeta_\tau, \quad l \in L, \tau \in \tilde{L}_l, j \in L \cup J. \tag{112b}$$

The flow balance constraints in $\tilde{G}$ are

$$\sum_{i \in I \cup L} y_{i\tau} = \sum_{j \in L \cup J} y_{\tau j}, \quad l \in L, \tau \in \tilde{L}_l \tag{113a}$$

$$y_{il} = \sum_{\tau \in \tilde{L}_l} y_{i\tau}, \quad l \in L, i \in I \cup L \tag{113b}$$

$$y_{lj} = \sum_{\tau \in \tilde{L}_l} y_{\tau j}, \quad l \in L, j \in L \cup J. \tag{113c}$$

The commodity balance constraints (8) are discretized as

$$y_{il} + \sum_{\substack{l' \in L: \\ i \in I_{l'}}} \sum_{\tau' \in \tilde{L}_{l'}} \tilde{q}^{\tau'}_{il'} y_{\tau' l} = \tilde{q}^\tau_{il} \sum_{j \in L \cup J} y_{\tau j}, \quad l \in L, i \in I_l, \tau \in \tilde{L}_l. \tag{114}$$

The specification requirement constraint (9) at the outputs is discretized as

$$\sum_{i\in I}\lambda_{ik}y_{ij}+\sum_{l\in L}\sum_{i\in I_l}\lambda_{ik}\sum_{\tau\in\tilde{L}_l}\tilde{q}_{il}^{\tau}y_{\tau j} \;\leq\; \mu_{jk}^{\max}\sum_{i\in I\cup L}y_{ij},\quad j\in J,k\in K \tag{115a}$$

$$\sum_{i\in I}\lambda_{ik}y_{ij}+\sum_{l\in L}\sum_{i\in I_l}\lambda_{ik}\sum_{\tau\in\tilde{L}_l}\tilde{q}_{il}^{\tau}y_{\tau j} \;\geq\; \mu_{jk}^{\min}\sum_{i\in I\cup L}y_{ij},\quad j\in J,k\in K. \tag{115b}$$

Since the values of $\tilde{q}_{il}^{\tau}$ are known for all $l\in L, i\in I_l, \tau\in\tilde{L}_l$, (114) and (115) are linear constraints. Thus, we have the following MILP approximation.

$$\begin{aligned}\min_{y,\zeta}\quad & \sum_{(i,j)\in\mathcal{A}}c_{ij}y_{ij}\\ \text{s.t.}\quad & y\in\mathcal{F}, (112)-(115).\end{aligned} \tag{$\mathcal{E}(\mathbb{RPQ})$}$$

The minimization in $\mathcal{E}(\mathbb{RPQ})$ is over (y,ζ) and there are exponentially many ζ variables and constraints (112b) and (114).

4.3 Computational results

In this section we report computational results on several test instances of the pooling problem. Our purpose is to assess usefulness of the proposed discretization models. We solve each discretization model and the original (nondiscretized) pooling problem. Then we compare the best feasible solutions that are obtained after solving all these models. For ratio and spec discretization, we try different values for the level of discretization. In our experiments, we do not implement our discretization strategies as part of a node heuristic while solving the pooling problem to global optimality. Our goal is to evaluate which discretization strategy empirically seems to work best on the pooling problem.

4.3.1 Experimental setup

`Cplex` 12.2 [62] was used as the MILP solver and `BARON` 9.0.1 [93, 104] and `Couenne` 0.4 [23] were used to solve BLPs and MIBLPs. We note that `BARON` is a commercially licensed solver whereas `Couenne` is developed under the open source COIN-OR project `http://www.coin-or.org`. Both these global solvers require local NLP solvers for their heuristics. `SNOPT` 7.2 [48] was used as the NLP solver with `BARON` whereas `Ipopt` 3.8 [112] was used with `Couenne`. `Cplex` 12.2 was also used as the LP solver with `BARON` and `Couenne`. All

formulations were modeled using GAMS 23.6 [47]. The time limit for each solve was set to 1hr. We compare the best feasible solutions obtained by solving the different models. All experiments were run on a Linux machine with kernel 2.6.18 running on a 64-bit x86 processor and 32GB of RAM.

Given an instance of the pooling problem, we solve the two formulations $\mathbb{P}$ and $\mathbb{PQ}$ as BLPs for standard instances or as MIBLPs for general instances. We now discuss the various discretization models solved for $\mathbb{PQ}$ and comment that a similar technique can be extrapolated to $\mathbb{P}$. For $\mathbb{PQ}$, we solved $\mathbb{FPQ}$ and $\mathbb{RPQ}$ as MIBLPs, and $\mathcal{B}(\mathbb{FPQ})$, $\mathcal{B}(\mathbb{RPQ})$, $\mathcal{U}(\mathbb{RPQ})$, $\mathcal{L}(\mathbb{RPQ})$, and $\mathcal{E}(\mathbb{RPQ})$ as MILPs. For flow discretization, amongst all the MILPs, we considered only the binary expansion model $\mathcal{B}(\mathbb{FPQ})$ and discretized flows within their bounds, as explained in §4.1.1. In ratio discretization, we tested different values of n. Clearly as n increases, there is a tradeoff between finding good feasible solutions versus being unable to solve the model to optimality due to its large size. For $n \in \{1, 2, 4\}$, we solved $\mathbb{RPQ}$, $\mathcal{U}(\mathbb{RPQ})$, and $\mathcal{E}(\mathbb{RPQ})$. For higher values $n \in \{7, 15, 31\}$, we solved $\mathbb{RPQ}$, $\mathcal{U}(\mathbb{RPQ})$, $\mathcal{L}(\mathbb{RPQ})$, and $\mathcal{B}(\mathbb{RPQ})$. Since the number of variables and constraints in $\mathcal{E}(\mathbb{RPQ})$ grows exponentially with n, we did not consider this model for high values of n. The effect of the proposed valid inequalities from §4.1.1.1 was tested with both $\mathcal{B}(\mathbb{FPQ})$ and $\mathcal{B}(\mathbb{FP})$. These MILPs were implemented using ILOG Concert technology and inequalities were separated only at the root node. In our initial testing with ratio and spec discretization, we found no significant advantage of using these cuts with either $\mathcal{B}(\mathbb{RPQ})$ or $\mathcal{B}(\mathbb{SP})$.

To ensure numerical consistency among the different solvers, we used the following algorithmic parameters: `feasibility tolerance` = 10^{-6}, `integrality tolerance` = 10^{-5}, `relative optimality gap` = 0.01%, and `absolute optimality gap` = 10^{-3}. Additionally, for `Cplex`, we set `Threads = 1` and `MIPEmphasis = Feasibility`. The `MIPEmphasis` parameter is used to aid `Cplex` in finding good feasible solutions at the expense of proof of optimality. We do not know of a similar parameter for `BARON` and `Couenne`.

In all the experiments, we report the best feasible solution (if one exists) and the corresponding upper bound as returned by the solver at the end of 1hr. If no feasible solution is returned because the solver could not find one within 1hr, then we use the best upper bound

value that might be reported by the solver and mark it with †. If an upper bound value is not reported but the solver has not determined if the model is infeasible, we mark with a $-$. Otherwise the model is provably infeasible and the upper bound is $+\infty$. If a model is solved to within 0.01% optimality in 1hr, then the total solution time (in seconds) is noted in parenthesis. Otherwise, we report the % optimality gap upon termination. Optimality gap at termination is defined as

$$\% \text{ optgap } = 100 \times \left| 1 - \frac{\text{Best lower bound at termination}}{\text{Best upper bound at termination}} \right|.$$

4.3.2 Test instances

The pooling instances commonly used in literature mostly comprise the small-scale problems proposed many years ago [3, 25, 60]. Since these problems are solved in a matter of seconds by the current versions of `BARON` and `Couenne`, they are not of particular interest in testing our discretization methods. Our test set comprises of fifty two medium- and large-scale instances of the pooling problem. We explain these instances next.

Standard pooling. Thirty two instances of standard pooling problems are used in our experiments. Twenty of these were created by Alfaki and Haugland [6] and can be downloaded from `http://www.ii.uib.no/~mohammeda/spooling/`. These instances are labeled as `stdA0-stdA9, stdB0-stdB5, stdC0-stdC3`. Another twelve instances are used from Ruiz et al. [92], available at `http://www.g-scop.fr/~penzb`, and are labeled as `jogo.*`. We chose only those instances by Ruiz et al. that were difficult to solve with `BARON` or `Couenne` (more than 15mins). As mentioned in §1.4, these instances are a variant of the typical pooling problem. The total flow into an output, given by $\sum_{i \in I \cup L} y_{ij}$, is fixed to some positive demand, for each output $j \in J$. Also, nontrivial lower and upper bounds are imposed for flow ratios on each arc and the specification produced at each pool. Nonetheless, our discretization models can be easily adapted to this variant.

Generalized pooling. Of the twenty generalized pooling instances, three are by Meyer and Floudas [72], also available in [73]. In these instances, besides the classical pooling

problem of §1.2, there are additional binary decision variables related to the use of each arc or node in the graph. Also, the specification tracking constraints (4a) are formulated as

$$\eta_{lk}\left[\sum_{i\in I}\lambda_{ik}y_{il}+\sum_{l'\in L}p_{l'k}y_{l'l}\right] \;=\; p_{lk}\sum_{j\in L\cup J}y_{lj},\quad l\in L, k\in K,$$

where η_{lk} is an absorption coefficient of spec k at pool l. Hence, to write the pq-formulation of this problem, we need to define ratio variables along each path such that q_{il}^{τ} : ratio of incoming flow to l along path τ starting from input i, and express

$$p_{lk}=\sum_{i}\sum_{\tau}\lambda_{ik}q_{il}^{\tau}.$$

Although this can be done, it makes the formulation larger in size due to its path dependency and hence we do not consider ($\mathbb{PQ}$) and its discretizations for the three instances of Meyer and Floudas. Some instances of the generalized pooling problem can also be found in Alfaki and Haugland [7]. However, in our experience, the pq-formulations of these instances were solved by `BARON` in less than 15 minutes and hence we chose not to include them in our test set due to their relative ease. In order to further expand our test set, we generated 17 random instances, `Inst1 - Inst17`, of the time indexed pooling problem stated in §1.4.1.

Table 18 presents the sizes of the instances used in this study. In particular, we report the size of the graph on which the pooling problem is defined. The actual number of bilinear constraints and terms can then be derived from Table 1. For the random instances `Inst*`, the reported number of inputs, pools, and outputs, is equal to the length of time period multiplied by the original number of inputs, pools, and outputs, respectively, see the discussion in §1.4.1 for transforming these instances to a generalized pooling problem.

Based on the problem sizes from Table 18, we classify these fifty two instances into medium- and large-scale. For medium-scale, we include the following seventeen instances : `stdA*`, `meyer*`, and `Inst1-4`. The remaining thirty five instances are classified as large-scale.

Table 18: Characteristics of the pooling instances.

Type	Source	Num of	Label	Inputs $\|I\|$	Pools $\|L\|$	Outputs $\|J\|$	Specs $\|K\|$
Standard	[6]	10	`stdA*`	20	10	15	12
		6	`stdB*`	35	17	21	17
		4	`stdC*`	60	30	40	20
	[92]	12	`jogo.*`	60	$\{7, 9, 11\}$	$\{20, 25, 30, 35, 45\}$	$\{72, 75, 78, 80, 82\}$
General	[72]	3	`meyer*`	7	$\{4, 10, 15\}$	1	3
	§1.4.1	4	`Inst1-4`	$\{12, 18\}$	$\{30, 42, 48\}$	12	2
		9	`Inst5-11,15,16`	48	60	30	4
		4	`Inst12-14,17`	80	100	50	4

4.3.3 Preprocessing

We now mention a preprocessing technique, motivated by the structure of the pooling problem, for reducing the number of constraints in the formulation. Consider the the *pq*-formulation and recall the specification requirement constraint (9) for any $j \in J, k \in K$.

$$\mu_{jk}^{\min} \sum_{i \in I \cup L} y_{ij} \;\leq\; \sum_{i \in I} \lambda_{ik} y_{ij} + \sum_{l \in L} \sum_{i \in I_l} \lambda_{ik} q_{il} y_{lj} \;\leq\; \mu_{jk}^{\max} \sum_{i \in I \cup L} y_{ij}$$

Here q_{il} denotes the ratio of incoming flow to pool l that started at input i. If $\sum_{t \in I \cup L} y_{tj} > 0$, then scaling both sides of above by $\sum_{t \in I \cup L} y_{tj}$ gives the following interpretation of (9) :

$$\sum_{t \in I \cup L} y_{tj} > 0 \quad \Rightarrow \quad p_{j\cdot} \in \operatorname{conv}\left(\cup_{i \in I_j} \lambda_{i\cdot}\right), \;\; p_{j\cdot} \in [\mu_{j\cdot}^{\min}, \mu_{j\cdot}^{\max}],$$

where $p_{j\cdot}$ is the vector of concentration values produced at this output j. This simple observation motivates our preprocessing method. We want to check if $\sum_{t \in I \cup L} y_{tj} > 0$ implies

$$\operatorname{conv}\left(\cup_{i \in I_j} \lambda_{i\cdot}\right) \bigcap [\mu_{j\cdot}^{\min}, \mu_{j\cdot}^{\max}] \;\neq\; \emptyset.$$

This can be easily verified by solving an LP.

Table 19: Effects of LP preprocessing on standard pooling instances. For every instance, we report the number of deleted outputs ($j \in J$) and the number of deleted constraints of the type (5).

#	Deleted		#	Deleted		#	Deleted	
	$j \in J$	(5)		$j \in J$	(5)		$j \in J$	(5)
stdA0	3	20	stdB1	3	9	jogo.21	0	889
stdA1	5	21	stdB2	0	15	jogo.22	0	953
stdA2	5	13	stdB3	4	15	jogo.23	0	1482
stdA3	3	15	stdB4	5	10	jogo.24	0	1236
stdA4	2	9	stdB5	2	8	jogo.25	0	2228
stdA5	4	16	stdC0	5	15	jogo.26	0	1046
stdA6	0	15	stdC1	4	18	jogo.27	0	1200
stdA7	3	6	stdC2	3	10	jogo.28	0	1231
stdA8	1	3	stdC3	7	13	jogo.29	0	1759
stdA9	6	11	jogo.15	0	909	jogo.30	0	2353
stdB0	5	15	jogo.17	0	1439			

Proposition 4.4 (LP preprocessing)**.** *For any $j \in J$ such that the system of linear inequalities*

$$\mu_{j\cdot}^{\min} \;\le\; \sum_{i \in I_j} \lambda_{i\cdot} \xi \;\le\; \mu_{j\cdot}^{\max}, \quad \sum_{i \in I_j} \xi_i = 1,\; \xi \ge \mathbf{0} \tag{116}$$

is infeasible, then $\sum_{t \in I \cup L} y_{tj} = 0$ is valid to the pooling problem and hence output node j can be deleted from the graph.

If (116) *is feasible, then for any $k \in K$ such that*

$$\max\{\lambda_{ik} \colon i \in I_j\} \;\le\; \mu_{jk}^{\max}, \quad \text{or} \quad \min\{\lambda_{ik} \colon i \in I_j\} \;\ge\; \mu_{jk}^{\min}$$

we can relax (5a) *or* (5b)*, respectively.*

Relaxing a constraint from (5) also implies that we relax the corresponding constraint in (9) for the *pq*-formulation.

We report the number of deleted output nodes and specification requirement constraints in Table 19. Preprocessing did not have any effect on the generalized problems.

4.3.4 Global optimal solutions

Here, we compare the performance of the two global solvers, `BARON` and `Couenne`, on our test instances. We provide the best upper bounds obtained with each of these two solvers. We also compare the effect of branching on bilinear terms. This can be explained as follows. While formulating a bilinear program, one may replace every occurrence of a bilinear term $\chi\rho$ with an auxiliary variable w and then explicitly add the defining constraints $w = \chi\rho$ to the formulation. In theory, this is still an equivalent problem. However, the global solver behaves differently under this replacement. For example, the solver now detects w to be an original problem variable and $w = \chi\rho$ to be just another bilinear constraint. Hence, the solver will now preprocess, bound tighten, and branch on w, in addition to χ and ρ, in its branch-and-bound algorithm. Branching on the bilinear term $\chi\rho$ is optional in the original formulation where w is absent and may or may not occur at all nodes of the search tree. Additionally, for the original formulation without w, the automatic cut generators of the solver may separate inequalities obtained from multiterm relaxations [cf. 18]. Such a pre-emption of cut generation after introducing w usually does not occur in `Couenne` [22].

We also remark here that for the equality constraints (4a) and (8), the bilinear term on the right hand side can either be aggregated or disaggregated, each representation leading to a different relaxation using McCormick envelopes. In general, the strengths of these two relaxations do not compare to each other, as was shown in Proposition 2.4. The disaggregated terms are necessary since they also appear in (5) ((9) for $\mathbb{PQ}$). We initially tested introducing auxiliary variables for the aggregated term also but found no significant benefit in solving the problem to optimality. Hence, we assume disaggregated representation of the right hand side of (4a) and (8) for the rest of the chapter.

The results of the global solve are presented in Table 20. Solution values are rounded to the nearest integer. For both `BARON` and `Couenne`, we compare the effect of explicitly reformulating each bilinear term $\chi\rho$ using a new variable $w = \chi\rho$. Both the formulations ($\mathbb{P}$) and ($\mathbb{PQ}$) were tested. However only one set of best upper bound values upon termination is provided. We observed that for the standard instances the ($\mathbb{PQ}$) formulation performed vastly better than the ($\mathbb{P}$) formulation. This is perhaps to be expected since, as proved in

§1.5.2.1, ($\mathbb{PQ}$) admits a stronger relaxation than ($\mathbb{P}$) for any pooling instance. However, for the generalized instances from our test set, the stronger relaxation of ($\mathbb{PQ}$) did not lead to a improved branch-and-bound performance. We can explain this as follows. First of all, as noted in §4.3.2, we did not consider ($\mathbb{PQ}$) for the `meyer*` instances due to the path dependency of the ratio variables and hence its extremely large size. Our randomly generated instances `Inst*` can be transformed to a generalized pooling instance as explained in §1.4.1. This transformation creates, amongst other things, multiple copies of each input at each time. Then, at time t, any pool receives flows from any input corresponding to an earlier time $t' \leq t$. Hence there are a large number of ratio variables in the ($\mathbb{PQ}$) formulation, which may explain its slower performance than ($\mathbb{P}$). Hence, in Table 20, we present only best upper bound values from ($\mathbb{P}$) for the generalized pooling instances.

From Table 20, we observe that `BARON` provides better quality solutions for the standard instances in our test set. On the generalized time indexed instances, `Couenne` was able to find feasible solutions on 5 instances, whereas `BARON` always provided some finite upper bound without finding a feasible solution. The effect of introducing $w = \chi\rho$ is mixed. It does not seem very useful on medium-scale instances. It helps on a few large-scale instances such as `stdB0-B5,Inst7`, but the model becomes too large and hence is slow in the case of `stdC0-C3,jogo.28-30`.

4.3.5 MILP results

In this section, we present computational results from solving the different MILP discretizations. Each MILP model of §4.1 and §4.2, if feasible, provides a feasible solution to the pooling instance. Hence, if we simply run the MILP model for 1hr, we will get a upper bound on the pooling instance, assuming a feasible solution is found to the MILP. We also perform an additional step to improve this upper bound. We use the feasible solution obtained by solving the MILP to warm-start `SNOPT`, a NLP solver. The NLP solver will perform a local search around the incumbent solution and try to find a better local minimizer. Hence, this may lead to an improved upper bound to the original problem. The generalized instances in our test set are mixed integer bilinear programs and in this case while warm-starting the

Table 20: Global optimal solutions for 52 pooling instances using BARON and Couenne. Two techniques are tested: each bilinear term $\chi\rho$ is either retained as is (No $w = \chi\rho$) or reformulated using a new variable $w = \chi\rho$. For standard instances, values are from ($\mathbb{PQ}$) whereas for general instances, values are from ($\mathbb{P}$). * means instance solved to optimality, in which case total time is mentioned in parenthesis. Otherwise, some upper bound is available and % optimality gap is provided. † means upper bound does not correspond to a feasible solution. - means no finite upper bound is calculated but model has not been proven infeasible either.

STANDARD : values from solving ($\mathbb{PQ}$)					GENERAL : values from solving ($\mathbb{P}$)				
#	BARON		Couenne		#	BARON		Couenne	
	$w = \chi\rho$	No $w = \chi\rho$	$w = \chi\rho$	No $w = \chi\rho$		$w = \chi\rho$	No $w = \chi\rho$	$w = \chi\rho$	No $w = \chi\rho$
stdA0	**-35812** (4%)	**-35812** (4%)	-35727 (3%)	-35138 (7%)	meyer4	1.116e6 (3%)	**1.086e6*** (3383)	1.246e6 (34%)	
stdA1	**-29277** (3%)	**-29277** (3%)	-29240 (4%)	-29207 (3%)	meyer10	1.146e6 (29%)	**1.124e6** (26%)	1.349e6 (46%)	
stdA2	**-23044*** (483)	**-23044*** (459)	-22985 (4%)	**-23044*** (3138)	meyer15	9.845e5 (31%)	**9.660e5** (27%)	–	–
stdA3	**-39447** (3%)	**-39447** (3%)	-38493 (9%)	-39411 (2%)	Inst1	22383† (91%)		**2365** (11%)	2370 (13%)
stdA4	**-41257** (4%)	**-41257** (4%)	-39856 (9%)	-40697 (5%)	Inst2	32534† (93%)		**2639** (15%)	**2639** (15%)
stdA5	**-27901** (1%)	**-27901** (1%)	-26547 (6%)	-27775 (2%)	Inst3	2891 (6%)	2872 (5%)	2797 (2%)	**2738** (0.03%)
stdA6	**-42138** (0.77%)	-42077 (0.91%)	0	-42072 (0.93%)	Inst4	2433 (3%)	2454 (4%)	2407 (2%)	**2396** (0.84%)
stdA7	-44382 (0.68%)	**-44400** (0.64%)	0	-43760 (2%)	Inst5	70329† (81%)	**70301**† (81%)	–	–
stdA8	-30532 (0.44%)	**-30556** (0.36%)	0	-30436 (0.76%)	Inst6	68743† (80%)	**68507**† (80%)	–	–
stdA9	-21914 (0.09%)	**-21933*** (3084)	0	-21912 (0.10%)	Inst7	**55350**† (73%)	66826† (79%)	–	–
stdB0	**-42466** (7%)	-35848 (26%)	0	-42352 (7%)	Inst8	57855† (83%)		**11139** (14%)	–
stdB1	**-63361** (3%)	-56201 (16%)	0	0	Inst9	**56280**† (84%)		–	–
stdB2	**-46899** (20%)	-5126 (999%)	0	0	Inst10	47248† (71%)		–	**15374** (12%)
stdB3	**-65750** (13%)	-11703 (533%)	0	0	Inst11	**48260**† (70%)		–	–
stdB4	**-59365** (0.18%)	-12168 (389%)	0	0	Inst12	**84363**† (68%)		–	–
stdB5	**-4711** (1188%)	-4261 (1324%)	0	0	Inst13	**98375**† (60%)		–	–
stdC0	-30964 (217%)	**-68325** (44%)	0	0	Inst14	**86095**† (73%)		–	–
stdC1	-7307 (1525%)	**-14941** (694%)	0	0	Inst15	**47181**† (73%)		–	–
stdC2	-5830 (2240%)	**-12632** (975%)	0	0	Inst16	**48843**† (75%)		–	–
stdC3	-4214 (2992%)	**-8193** (1491%)	0	0	Inst17	**84126**† (60%)		–	–
jogo.15	**1.7165e6** (0.69%)	1.7338e6 (2%)	–	1.7171e6 (2%)					
jogo.17	**2.5811e6** (0.81%)	2.5815e6 (1%)	–	–					
jogo.21	**2.1618e6** (0.61%)	**2.1618e6** (0.50%)	–	3.7090e6 (43%)					
jogo.22	**1.6286e6** (3%)	1.6356e6 (2%)	–	1.6422e6 (4%)					
jogo.23	**2.1269e6** (2%)	2.1270e6 (1%)	–	2.1330e6 (2%)					
jogo.24	**1.3562e6** (2%)	**1.3562e6** (2%)	–	–					
jogo.25	3.3694e6 (2%)	**3.3466e6** (2%)	–	–					
jogo.26	**2.3346e6*** (1523)	**2.3346e6*** (199)	–	2.3397e6 (2%)					
jogo.27	**2.2690e6** (2%)	2.2705e6 (2%)	–	–					
jogo.28	–	**1.6149e7** (77%)	–	–					
jogo.29	–	**3.8456e6** (4%)	–	–					
jogo.30	–	–	–	–					

BLP, we fixed the integer variables to values obtained from the MILP solution and solved the resulting partially fixed BLP. Upper bounds from both the MILP discretization and the warm-started BLP are reported in the subsequent tables. The BLP was allowed to run for 10 minutes. Since the MILP was run for 1hr with `Cplex`, the total time for our heuristic method becomes 70 minutes, greater than the 60 minutes for which a global solver was run. However, after inspecting the output from the global solver, we believe that additional 10 minutes are not likely to substantially increase the solution quality obtained from the global solver. Also, on most instances, the MILP solution after 1hr is already a tighter upper bound than that due to the global solver.

Flow discretization. We first consider the flow discretization model explained in §4.1.1. Discretizations for both ($\mathbb{P}$) and ($\mathbb{PQ}$) formulations were tested. For both ($\mathbb{P}$) and ($\mathbb{PQ}$), the flow discretized model $\mathbb{FP}$ and $\mathbb{FPQ}$, respectively, was solved as a MIBLP using `BARON` and `Couenne`. We report only the best of the two values produced by these two MINLP solvers. As observed during the global solve, ($\mathbb{PQ}$) and ($\mathbb{P}$) performed better on standard and generalized instances, respectively. The results are presented in Table 21. Numbers in parenthesis are either solution time in seconds or % optimality gap at termination. Note that this optimality gap is calculated with reference to the best lower bound provided by the respective solver. Upper bounds obtained by warm-starting BLP with MILP solution are reported in the last column (MILP + BLP w.s.). The best discretized solution, without considering the solution obtained by warm-starting BLP, is highlighted in bold.

We make the following observations from Table 21. The MILP values are almost always better than solving a MIBLP (except for `stdC0`). MILP was solved using two approaches: either default `Cplex` settings (NoCuts) or default `Cplex` and cutting planes of (101) and §4.1.1.1 (Cuts). The addition of cuts had almost negligible effect. The average root gap closed was very low, around 1-2% (not reported in table). MILP solved with cuts sometimes produced a better quality feasible solution, such as for `stdA0-A5`. However on most of the other instances, the default `Cplex` settings performed equally well or better. For warm-starting the BLP, we always used the feasible solution provided by MILP solved without

our cuts. For standard problems, the warm-started BLP produced the best upper bound on all but two instances. For the generalized time indexed problems, warm-starting by fixing the original integer variables seemed to have no effect on the MILP solution.

Ratio and specification discretization. We now discuss the discretization of non-flow variables in the formulation. Discretizing specifications in ($\mathbb{P}$) was not found to be a very useful approach. It always produced inferior quality solutions than those obtained from flow discretization in ($\mathbb{P}$). Henceforth, we do not consider the MILP model ($\mathbb{SP}$). Ratio discretization produced some encouraging results. In particular, for many of the standard `std*` instances, it produced better upper bounds than flow discretization in ($\mathbb{PQ}$). For the `jogo*` instances though, the ratio MILPs, $\mathcal{B}(\mathbb{RPQ})$,$\mathcal{U}(\mathbb{RPQ})$, and $\mathcal{L}(\mathbb{RPQ})$, were either provably infeasible or `Cplex` was unable to find a feasible solution after 1hr. Similarly for the `meyer*` instances. On some of the time indexed `Inst*` instances, feasible solutions were found for small values of n, i.e. $n \in \{1, 2, 4\}$. We also considered the expanded network MILP $\mathcal{E}(\mathbb{RPQ})$ explained in §4.2.2, since it is another reformulation of $\mathbb{RPQ}$. This formulation was tested only for $n \in \{1, 2, 4\}$ due to its exponentially many constraints and variables. It produced good quality solutions only for the large-scale `std*` instances for $n \in \{1, 2\}$.

The results are reported in Table 22. For each instance, the best upper bound obtained by ratio discretization and the corresponding model and value of n are presented. As before, numbers in parenthesis are either solution time in seconds or % optimality gap at termination. The number in square bracket represents the value of n, the level of discretization, for which the corresponding model produced the best solution. BLP was warm-started using the best MILP solution. If two different values of n produced the same best solution, then we only report the smaller value of n. In case of a tie-break between two MILPs for the best solution, the priority in decreasing order is $\mathcal{B}(\mathbb{RPQ})$,$\mathcal{U}(\mathbb{RPQ})$, $\mathcal{L}(\mathbb{RPQ})$, $\mathcal{E}(\mathbb{RPQ})$.

From Table 22, we observe that there is no one MILP model that works best on all the instances. Also, the level of discretization is not unique. For standard and general medium-scale instances, binary reformulation of $\mathbb{RPQ}$ works quite well for higher values

Table 21: Feasible solutions by discretizing outflows from each pool. Best MILP solution is highlighted in bold for each instance. BLP is warm-started using solution from MILP without user cuts.

STANDARD : values from solving $\mathbb{FPQ}$					GENERAL : values from solving $\mathbb{FP}$				
#	MIBLP	MILP $\mathcal{B}(\mathbb{FPQ})$		MILP +	#	MIBLP	MILP $\mathcal{B}(\mathbb{FP})$		MILP +
	$\mathbb{FPQ}$	NoCuts	Cuts	BLP w.s.		$\mathbb{FP}$	NoCuts	Cuts	BLP w.s.
stdA0	-35732 (3%)	-35777 (2%)	**-35794** (2%)	-35812	meyer4	3.04e6† (64%)	**1.11e6*** (1887)	1.11e6* (2013)	1.11e6
stdA1	-28974 (6%)	-29224 (2%)	**-29269** (2%)	-29277	meyer10	1.13e7† (94%)	**1.50e6** (49%)	1.67e6 (55%)	1.49e6
stdA2	-22984 (1%)	**-23037*** (621)	-23037 (0.10%)	-23044	meyer15	17.23e6† (96%)	**1.07e6** (31%)	1.08e6 (33%)	1.05e6
stdA3	-38574 (7%)	-39430 (2%)	**-39436** (2%)	-39446	Inst1	22383† (91%)	**2296*** (1055)	2349 (7%)	2296
stdA4	-40015 (8%)	-40506 (6%)	**-40928** (5%)	-40692	Inst2	32534† (93%)	**2428** (2%)	2586 (10%)	2428
stdA5	**-27421** (3%)	-26774 (6%)	-27347 (3%)	-27567	Inst3	2738* (1123)	**2738*** (35)	2738* (257)	2738
stdA6	-41874 (1%)	**-41890** (1%)	-41825 (2%)	-42095	Inst4	37382† (94%)	**2395*** (149)	2395* (156)	2395
stdA7	-43568 (3%)	**-44020** (1%)	-43537 (3%)	-44240	Inst5	70301† (98%)	**15217** (11%)	15230 (11%)	15217
stdA8	-30155 (2%)	**-30478** (0.62%)	-30447 (0.72%)	-30514	Inst6	**68508**† (80%)	–	–	–
stdA9	-21823 (0.51%)	-21817 (0.53%)	**-21917** (0.08%)	-21856	Inst7	**66827**† (79%)	–	–	–
stdB0	-41636 (9%)	-41076 (11%)	-41844 (9%)	-42814	Inst8	57855† (83%)	**11456** (13%)	11456 (13%)	11456
stdB1	**-62704** (4%)	-62588 (4%)	-62339 (5%)	-63200	Inst9	56280† (84%)	**9175** (0.06%)	9214 (0.35%)	9175
stdB2	-47831 (18%)	-49589 (14%)	**-52051** (8%)	-54048	Inst10	47248† (71%)	**14681** (6%)	**14681** (6%)	14681
stdB3	**-72774** (2%)	-71496 (4%)	-57668 (28%)	-73841	Inst11	48260† (70%)	**15316** (3%)	15420 (5%)	15316
stdB4	-1836 (3140%)	**-28189** (115%)	0	-55246	Inst12	**84363**† (68%)	–	–	–
stdB5	0	**-30196** (104%)	0	-40648	Inst13	**98375**† (60%)	–	–	–
stdC0	**-66026** (49%)	-44033 (124%)	-45045 (119%)	-59990	Inst14	**86095**† (73%)	–	–	–
stdC1	-8178 (1353%)	**-10999** (988%)	0	-11006	Inst15	47926† (70%)	**13359** (3%)	13728 (12%)	13359
stdC2	-13274 (924%)	**-20871** (554%)	0	-28890	Inst16	48843† (75%)	**12956** (5%)	13014 (7%)	12956
stdC3	-1038 (12450%)	**-28620** (372%)	0	-52720	Inst17	**84126**† (60%)	–	–	–
jogo.15	6.5797e6† (74%)	**1.7181e6** (1%)	1.7193 (1%)	1.7165e6					
jogo.17	7.9053e6† (68%)	**2.5828e6** (2%)	2.9715e6 (4%)	2.5810e6					
jogo.21	10.7991e6† (80%)	**2.1624e6** (0.35%)	3.2780e6 (9%)	2.1617e6					
jogo.22	10.9354e6† (85%)	**1.6307e6** (2%)	2.3544e6 (11%)	1.6277e6					
jogo.23	16.1219e6† (87%)	**2.1287e6** (1%)	2.1349e6 (5%)	2.1287e6					
jogo.24	10.1511e6† (87%)	**1.3552e6** (2%)	1.3560e6 (2%)	1.3543e6					
jogo.25	21.6524e6† (85%)	**3.3600e6** (3%)	3.3729e6 (5%)	3.3443e6					
jogo.26	2.3484e6 (0.28%)	**2.3386e6** (1%)	2.3391e6 (2%)	2.3346e6					
jogo.27	15.1121e6† (85%)	**2.2739e6** (3%)	2.2798e6 (6%)	2.2711e6					
jogo.28	16.1015e6† (77%)	**3.8374e6** (6%)	3.8399e6 (6%)	3.7904e6					
jogo.29	21.7426e6† (83%)	**3.8212e6** (2%)	3.8212e6 (2%)	3.8141e6					
jogo.30	19.8672e6† (80%)	**4.5578e6** (14%)	5.3312e6 (21%)	4.3224e6					

Table 22: Feasible solutions by discretizing inflow ratios in ($\mathbb{PQ}$). Best MILP upper bound and corresponding model presented for each instance. `jogo*` and remaining `Inst*` instances were either provably infeasible for small n or `Cplex` failed to find any solution within 1hr. ($\mathbb{PQ}$) not applicable to `meyer*` instances. Numbers in square bracket are values of n, the level of discretization.

Type	#	MIBLP	MILP		MILP + BLP
		$\mathbb{RPQ}$	Value	Model	warm-start
Standard medium-scale	stdA0	-33979 (8%) [7]	**-35735** (2%)	$\mathcal{B}(\mathbb{RPQ})$ [31]	-35812
	stdA1	-28785 (4%) [31]	**-29024** (2%)	$\mathcal{B}(\mathbb{RPQ})$ [31]	-29240
	stdA2	-22963 (1%) [31]	**-23004*** (1103)	$\mathcal{B}(\mathbb{RPQ})$ [31]	-23044
	stdA3	-38404 (5%) [15]	**-39348** (1%)	$\mathcal{B}(\mathbb{RPQ})$ [31]	-39446
	stdA4	-38394 (11%) [7]	**-40981** (3%)	$\mathcal{B}(\mathbb{RPQ})$ [15]	-41127
	stdA5	**-27726** (2%) [31]	-27471 (3%)	$\mathcal{B}(\mathbb{RPQ})$ [15]	-27717
	stdA6	**-42163** (1%) [31]	-42076 (1%)	$\mathcal{B}(\mathbb{RPQ})$ [15]	-42129
	stdA7	**-44251** (1%) [31]	-44086 (1%)	$\mathcal{B}(\mathbb{RPQ})$ [15]	-44332
	stdA8	-30448 (1%) [31]	**-30570** (0.32%)	$\mathcal{B}(\mathbb{RPQ})$ [7]	-30613
	stdA9	-21876 (0.26%) [15]	**-21889** (0.21%)	$\mathcal{E}(\mathbb{RPQ})$ [2]	-21889
Standard large-scale	stdB0	-17255 (163%) [4]	**-42486** (7%)	$\mathcal{B}(\mathbb{RPQ})$ [15]	-42870
	stdB1	-5075 (1187%) [2]	**-62829** (3%)	$\mathcal{B}(\mathbb{RPQ})$ [7]	-63214
	stdB2	-45224 (24%) [1]	**-53898** (4%)	$\mathcal{E}(\mathbb{RPQ})$ [2]	-54465
	stdB3	-8903 (732%) [2]	**-73677** (0.51%)	$\mathcal{E}(\mathbb{RPQ})$ [1]	-73939
	stdB4	-16535 (260%) [31]	**-59384** (0.14%)	$\mathcal{E}(\mathbb{RPQ})$ [1]	-59416
	stdB5	-7030 (763%) [2]	**-60012** (1%)	$\mathcal{E}(\mathbb{RPQ})$ [1]	-60377
	stdC0	-3335 (2842%) [2]	**-81112** (20%)	$\mathcal{B}(\mathbb{RPQ})$ [7]	-85167
	stdC1	-2459 (4736%) [2]	**-94997** (23%)	$\mathcal{E}(\mathbb{RPQ})$ [2]	-100156
	stdC2	-5854 (2230%) [2]	**-118948** (12%)	$\mathcal{E}(\mathbb{RPQ})$ [1]	-122687
	stdC3	-5214 (2399%) [2]	**-121368** (7%)	$\mathcal{E}(\mathbb{RPQ})$ [1]	-125238
General medium-scale	Inst1	2.24e4† (91%) [1]	**2382** (7%)	$\mathcal{B}(\mathbb{RPQ})$ [7]	2379
	Inst2	3.25e4† (93%) [1]	**2478** (5%)	$\mathcal{B}(\mathbb{RPQ})$ [7]	2455
	Inst3	1.85e4† (85%) [1]	**2753*** (9)	$\mathcal{U}(\mathbb{RPQ})$ [2]	2753
	Inst4	3.74e4† (94%) [1]	**2411** (1%)	$\mathcal{B}(\mathbb{RPQ})$ [7]	2411

of n. For large-scale problems, the expanded network MILP $\mathcal{E}(\mathbb{RPQ})$ dominates all other discretizations. However, this model gives good solutions only for small values of $n = 1, 2$. Note that $n = 1$ implies that there is no mixing at pools. Hence, this model $\mathcal{E}(\mathbb{RPQ})$ may not work well always. On almost all the instances, we get good solutions from warm-starting the bilinear program.

Outflow ratio discretization. Finally, we consider discretizing outflow ratios from each pool. This MILP model was explained in §4.2.1. Discretization of ($\mathbb{PQ}$) was superior for

standard problems whereas discretization of ($\mathbb{P}$) was superior on the generalized problems. We tested $n = 1, \ldots, 5$ for uniform model and $n = 2, \ldots, 6$ for asymmetric model. Results are presented in Table 23. The `jogo*` instances were either infeasible (for $n = 1$) or `Cplex` was unable to find any solution in 1hr for higher values of n. BLP was warm-started using better of the two solutions from uniform and asymmetric model.

On many of the instances, asymmetric discretization produced better quality feasible solutions. Although the uniform and asymmetric models are equivalent for $n = 2$ (and also $n = 1$), we may get different feasible solutions after 1hr if `Cplex` has not terminated. For the `meyer*` instances, the warm-started BLP found the best known global solutions (available in Misener and Floudas [73]). None of the discretizations produced a feasible solution for `Inst12`. As seen from the table, there is no clear choice for a suitable value of n. For `meyer4` and `meyer10`, SOS-1 outflows from each pool, i.e. $n = 1$ for $\mathbb{UEP}$, found nearly global solutions in a very short time. Similarly, $n = 1$ worked quite well for the large-scale standard instances `stdB4,stdB5,stdC1-C3`. For the time indexed problems `Inst*`, higher values $n \geq 3$ were required for most of the instances.

4.4 Summary

Table 24 presents a summary of our computational experiments. For each instance, we record the best upper bound value, denoted as $BestUB$, from global solution and amongst all the MILP discretizations. Note that the MILP value is the one obtained after warm-starting the BLP with a MILP solution. Our goal is to compare the quality of the solutions obtained by the different discretization methods. The metric that we use is the percentage gap between best upper bound and best lower bound. This gives us an estimate of how well a particular discretization model might perform if implemented as a heuristic in a branch-and-bound algorithm. Hence, we also report percentage gaps defined as

$$\% \text{ gap} = 100 \times \left| 1 - \frac{BestLB}{BestUB} \right|$$

for the best global solution from Table 20 and the best MILP bound. The best lower bound value ($BestLB$) is obtained from the global solution strategy, i.e. solver either `BARON` or `Couenne` and $w = \chi\rho$ or no $w = \chi\rho$, that yielded the best upper bound in Table 20. The

Table 23: Feasible solutions by discretizing consistency requirements at each pool. This can also be viewed as discretizing outflow ratios from each pool. Comparing two types of MILPs, uniform and asymmetric discretization, for each instance. Numbers in square bracket are values of n, the level of discretization.

Type and formulation	#	Uniform $\mathbb{UEP}$ or $\mathbb{UEPQ}$	Asymmetric $\mathbb{AEP}$ or $\mathbb{AEPQ}$	MILP + BLP warm-start
Standard medium-scale ($\mathbb{PQ}$)	stdA0	-35552 (1%) [5]	**-35665** (1%) [6]	-35812
	stdA1	-28721* (144) [5]	**-28917*** (217) [6]	-29085
	stdA2	-22741* (70) [5]	**-22780*** (205) [6]	-23042
	stdA3	-39095* (38) [3]	**-39154** (1%) [6]	-39383
	stdA4	-40791 (4%) [5]	**-40919** (3%) [5]	-41257
	stdA5	**-27113** (4%) [3]	-27070 (4%) [4]	-27368
	stdA6	-41980 (1%) [5]	**-41994** (1%) [5]	-42099
	stdA7	**-44335** (1%) [5]	-44317 (1%) [6]	-44368
	stdA8	-30504 (1%) [4]	**-30519** (0.49%) [5]	-30531
	stdA9	-21879 (0.25%) [3]	**-21916** (0.08%) [6]	-21920
Standard large-scale ($\mathbb{PQ}$)	stdB0	-42813 (5%) [4]	**-42887** (4%) [4]	-43097
	stdB1	-63009 (3%) [5]	**-63030** (2%) [5]	-63389
	stdB2	-53337 (5%) [5]	**-53360** (4%) [3]	-53873
	stdB3	**-73834** (0.29%) [3]	-73778 (0.33%) [2]	-73839
	stdB4	**-59445** (0.04%) [1]	-59317 (0.26%) [2]	-59451
	stdB5	**-60488** (0.34%) [1]	-60140 (1%) [5]	-60655
	stdC0	-82270 (13%) [3]	**-82790** (14%) [3]	-86082
	stdC1	**-100935** (10%) [1]	-99979 (16%) [2]	-103189
	stdC2	**-119529** (10%) [1]	-118080 (14%) [2]	-121651
	stdC3	**-124531** (5%) [1]	-121327 (7%) [2]	-124697
General medium-scale ($\mathbb{P}$)	meyer4	**1.087e6*** (2) [1]	1.087e6* (5) [2]	1.086e6
	meyer10	**1.087e6*** (69) [1]	1.087e6* (187) [2]	1.086e6
	meyer15	**945565** (4%) [5]	955529* (1366) [2]	943734
	Inst1	2358* (1773) [4]	**2300*** (484) [6]	2295
	Inst2	2451 (0.87%) [5]	**2442** (2%) [6]	2442
	Inst3	2774* (305) [5]	**2743*** (28) [6]	2738
	Inst4	**2395*** (100) [5]	2395* (116) [6]	2395
General large-scale ($\mathbb{P}$)	Inst5	14272 (4%) [5]	**14238** (4%) [4]	14179
	Inst6	14754 (4%) [2]	**14699** (4%) [4]	14699
	Inst7	14515 (3%) [3]	**14495** (4%) [6]	14488
	Inst8	10699 (6%) [5]	**10542** (3%) [4]	10420
	Inst9	9187 (0.24%) [4]	**9178** (0.09%) [6]	9175
	Inst10	14365* (3111) [3]	**14331** (1%) [5]	14314
	Inst11	15207 (0.31%) [4]	**15135** (0.75%) [6]	15119
	Inst12	–	–	–
	Inst13	41576 (4%) [2]	**41410** (3%) [2]	41265
	Inst14	**25523** (9%) [2]	–	25323
	Inst15	13354 (2%) [4]	**13348** (3%) [6]	13343
	Inst16	12897 (3%) [4]	**12647*** (3404) [5]	12647
	Inst17	**36081** (5%) [2]	36437 (6%) [2]	35837

best upper bound values are marked in bold, unless they are the same from the global solver and MILP discretization. The last column notes the MILP model and corresponding value of n in square brackets, if applicable, that yielded the best MILP solution.

The following observations can be made from Table 24. Flow discretization was the only MILP that yielded good solutions to the `jogo*` instances. These solutions were on most occasions, as good as or slightly better than those obtained from a global solver. For example, on `jogo28` and `jogo30`, the flow MILP substantially reduced the optimality gap to the lower bound. On the other hand, for `jogo26`, `BARON` was able to find the global solution faster. For the remaining standard instances `std*`, inflow or outflow ratio discretizations were generally a better choice than discretizing flows. There seems to be no clear choice for a fine enough level of discretization n. Note that for `stdB3`, `stdC2`, and `stdC3`, $n = 1$ worked well with $\mathcal{E}(\mathbb{RPQ})$, meaning that only one inflow at each pool and hence no pooling was required. On the generalized problems, outflow discretization was the overwhelming choice with no clear distinctions between the uniform and asymmetric models. The MILP approach is particularly helpful on the time indexed instances, since the optimality gaps are relatively small ($\leq 10\%$) and global solvers mostly produced finite upper bounds without finding a feasible solution. This perhaps lends credence to our original intuition that in the presence of combinatorial constraints, the MIBLP formulation of a pooling problem can be solved faster using some MILP techniques.

Thus, in this chapter, we have discussed different discretization methods to approximate a pooling problem as a MILP. We computationally tested these ideas on a set of 52 instances. Our experiments suggest that discretization seems to be a promising approach especially for large-scale and generalized pooling problems.

Table 24: Summary of discretization methods. Comparing best solution values and percentage gaps given as % gap $= 100 \times |1 - BestLB/BestUB|$, where $BestLB$ is obtained after 1hr of global solve.

STANDARD : values from $\mathbb{PQ}$				GENERAL : values from $\mathbb{P}$			
#	Global	MILP		#	Global	MILP	
	$BestUB$	$BestUB$	Model		$BestUB$	$BestUB$	Model
stdA0	-35812 (3.75%)	-35812 (3.75%)	$\mathcal{B}(\mathbb{FPQ})$	meyer4	1.0861e6* (3383)	1.0861e6* (2)	$\mathbb{UEP}$ [1]
stdA1	-29277 (2.79%)	-29277 (2.79%)	$\mathcal{B}(\mathbb{FPQ})$	meyer10	1.1242e6 (26.07%)	**1.0861e6** (23.48%)	$\mathbb{UEP}$ [1]
stdA2	-23044* (483)	-23044* (621)	$\mathcal{B}(\mathbb{FPQ})$	meyer15	965994 (26.75%)	**943734** (25.02%)	$\mathbb{UEP}$ [5]
stdA3	-39446 (2.73%)	-39446 (2.73%)	$\mathcal{B}(\mathbb{FPQ})$	Inst1	2365 (10.53%)	**2295** (7.80%)	$\mathbb{AEP}$ [6]
stdA4	-41257 (3.84%)	-41257 (3.84%)	$\mathbb{AEPQ}$ [5]	Inst2	2639 (15.01%)	**2428** (7.62%)	$\mathbb{AEP}$ [6]
stdA5	**-27901** (1.27%)	-27717 (1.94%)	$\mathcal{B}(\mathbb{RPQ})$ [15]	Inst3	2738 (0.04%)	2738 (0.04%)	$\mathbb{AEP}$ [6]
stdA6	-42138 (0.76%)	**-42129** (0.78%)	$\mathcal{B}(\mathbb{RPQ})$ [15]	Inst4	2396 (0.83%)	**2395** (0.79%)	$\mathbb{UEP}$ [5]
stdA7	**-44400** (0.64%)	-44368 (0.71%)	$\mathbb{AEPQ}$ [6]	Inst5	70301† (81.08%)	**14213** (6.42%)	$\mathbb{UEP}$ [5]
stdA8	-30556 (0.36%)	**-30613** (0.18%)	$\mathcal{B}(\mathbb{RPQ})$ [7]	Inst6	68507† (79.89%)	**14740** (6.53%)	$\mathbb{UEP}$ [2]
stdA9	**-21933***	-21920 (0.06%)	$\mathbb{AEPQ}$ [6]	Inst7	55350† (72.78%)	**14485** (4.03%)	$\mathbb{UEP}$ [3]
stdB0	-42466 (6.78%)	**-43097** (5.21%)	$\mathbb{AEPQ}$ [4]	Inst8	11139† (14.22%)	**10420** (8.30%)	$\mathbb{AEP}$ [4]
stdB1	-63361 (3.12%)	**-63389** (3.07%)	$\mathbb{AEPQ}$ [5]	Inst9	56280† (84.04%)	**9175** (2.08%)	$\mathbb{AEP}$ [6]
stdB2	-46899 (20.02%)	**-54465** (3.34%)	$\mathcal{E}(\mathbb{RPQ})$ [2]	Inst10	15374† (11.71%)	**14314** (5.18%)	$\mathbb{AEP}$ [5]
stdB3	-65750 (12.62%)	**-73939** (0.15%)	$\mathcal{E}(\mathbb{RPQ})$ [1]	Inst11	48259† (70.01%)	**15119** (4.26%)	$\mathbb{AEP}$ [6]
stdB4	-59365 (0.18%)	**-59451** (0.03%)	$\mathbb{UEPQ}$ [1]	Inst12	**84363**† (67.51%)	–	–
stdB5	-4711 (1188.39%)	**-60655** (0.07%)	$\mathbb{UEPQ}$ [1]	Inst13	983745† (60.21%)	**41265** (5.14%)	$\mathbb{AEP}$ [2]
stdC0	-68325 (43.56%)	**-86082** (13.94%)	$\mathbb{AEPQ}$ [3]	Inst14	86095† (73.18%)	**25323** (8.80%)	$\mathbb{UEP}$ [2]
stdC1	-14941 (694.49%)	**-103189** (15.04%)	$\mathbb{UEPQ}$ [1]	Inst15	47181† (73.51%)	**13343** (6.33%)	$\mathbb{AEP}$ [6]
stdC2	-12632 (974.94%)	**-122687** (10.68%)	$\mathcal{E}(\mathbb{RPQ})$ [1]	Inst16	48843† (75.01%)	**12647** (3.47%)	$\mathbb{AEP}$ [5]
stdC3	-8193 (1490.57%)	**-125238** (4.05%)	$\mathcal{E}(\mathbb{RPQ})$ [1]	Inst17	84126† (59.77%)	**35387** (4.37%)	$\mathbb{UEP}$ [2]
jogo.15	**1.7165e6** (0.69%)	1.7166e6 (0.70%)	$\mathcal{B}(\mathbb{FPQ})$				
jogo.17	2.5811e6 (0.82%)	**2.5810e6** (0.81%)	$\mathcal{B}(\mathbb{FPQ})$				
jogo.21	2.1618e6 (0.61%)	**2.1617e6** (0.61%)	$\mathcal{B}(\mathbb{FPQ})$				
jogo.22	1.6286e6 (2.66%)	**1.6277e6** (2.60%)	$\mathcal{B}(\mathbb{FPQ})$				
jogo.23	**2.1269e6** (1.50%)	2.1287e6 (1.59%)	$\mathcal{B}(\mathbb{FPQ})$				
jogo.24	1.3562e6 (1.90%)	**1.3543e6** (1.76%)	$\mathcal{B}(\mathbb{FPQ})$				
jogo.25	3.3466e6 (1.66%)	**3.3443e6** (1.59%)	$\mathcal{B}(\mathbb{FPQ})$				
jogo.26	**2.3346e6*** (1523)	2.3346e6 (3601)	$\mathcal{B}(\mathbb{FPQ})$				
jogo.27	**2.2690e6** (1.55%)	2.2711e6 (1.64%)	$\mathcal{B}(\mathbb{FPQ})$				
jogo.28	1.6149e7 (77.32%)	**3.7904e6** (3.36%)	$\mathcal{B}(\mathbb{FPQ})$				
jogo.29	3.8456e6 (3.80%)	**3.8141e6** (3.01%)	$\mathcal{B}(\mathbb{FPQ})$				
jogo.30	–	**4.3224e6** (18.17%)	$\mathcal{B}(\mathbb{FPQ})$				

REFERENCES

[1] K. Abhishek, S. Leyffer, and J. Linderoth. FilMINT: An outer approximation-based solver for convex mixed-integer nonlinear programs. *INFORMS Journal on Computing*, 22(4):555–567, 2010.

[2] W.P. Adams and H.D. Sherali. Mixed-integer bilinear programming problems. *Mathematical Programming*, 59(1):279–305, 1993.

[3] N. Adhya, M. Tawarmalani, and N.V. Sahinidis. A Lagrangian approach to the pooling problem. *Industrial and Engineering Chemistry Research*, 38(5):1956–1972, 1999.

[4] F. Al-Khayyal and S.J. Hwang. Inventory constrained maritime routing and scheduling for multi-commodity liquid bulk, Part I: Applications and model. *European Journal of Operational Research*, 176(1):106–130, 2007.

[5] F.A. Al-Khayyal and J.E. Falk. Jointly constrained biconvex programming. *Mathematics of Operations Research*, 8(2):273–286, 1983.

[6] M. Alfaki and D. Haugland. Strong formulations for the pooling problem. *Journal of Global Optimization*, pages 1–20, 2012.

[7] M. Alfaki and D. Haugland. A multi-commodity flow formulation for the generalized pooling problem. *Journal of Global Optimization*, pages 1–21, 2012.

[8] M. Alfaki and D. Haugland. Comparison of discrete and continuous models for the pooling problem. In A. Caprara and S. Kontogiannis, editors, *11th Workshop on Algorithmic Approaches for Transportation Modeling, Optimization, and Systems*, OpenAccess Series in Informatics, pages 112–121, 2011.

[9] H. Almutairi and S. Elhedhli. A new Lagrangean approach to the pooling problem. *Journal of Global Optimization*, 45(2):237–257, 2009.

[10] F. Amos, M. Ronnqvist, and G. Gill. Modelling the pooling problem at the New Zealand Refining Company. *Journal of the Operational Research Society*, 48(8):767–778, 1997.

[11] K.M. Anstreicher. On convex relaxations for quadratically constrained quadratic programming, 2010. URL `http://www.optimization-online.org/DB\_FILE/2010/08/2699.pdf`. Optimization Online, July 2012.

[12] A. Atamtürk and V. Narayanan. Polymatroids and mean-risk minimization in discrete optimization. *Operations Research Letters*, 36(5):618–622, 2008.

[13] A. Atamtürk and V. Narayanan. Lifting for conic mixed-integer programming. *Mathematical programming*, 126(2):351–363, 2011.

[14] C. Audet, J. Brimberg, P. Hansen, S. Le Digabel, and N. Mladenović. Pooling problem: Alternate formulations and solution methods. *Management Science*, 50(6):761–776, 2004.

[15] T.E. Baker and L.S. Lasdon. Successive linear programming at Exxon. *Management Science*, pages 264–274, 1985.

[16] E. Balas. Disjunctive programming: Properties of the convex hull of feasible points. *Discrete Applied Mathematics*, 89(1-3):3–44, 1998.

[17] E. Balas and R. Jeroslow. Canonical cuts on the unit hypercube. *SIAM Journal on Applied Mathematics*, 23(1):61–69, 1972.

[18] X. Bao, N.V. Sahinidis, and M. Tawarmalani. Multiterm polyhedral relaxations for nonconvex, quadratically constrained quadratic programs. *Optimization Methods and Software, 24*, 4(5):485–504, 2009.

[19] X. Bao, N.V. Sahinidis, and M. Tawarmalani. Semidefinite relaxations for quadratically constrained quadratic programming: A review and comparisons. *Mathematical Programming*, 129(1):129–157, 2011.

[20] P. Belotti. Disjunctive cuts for nonconvex MINLP. In J. Lee and S. Leyffer, editors, *Mixed Integer Nonlinear Programming*, volume 154 of *IMA Volumes in Mathematics and its Applications*, pages 117–144. Springer, 2012.

[21] P. Belotti. Couenne: a user's manual, 2012. URL `https://projects.coin-or.org/Couenne/browser/trunk/Couenne/doc/`. June 2012.

[22] P. Belotti. Private communication, March 2012.

[23] P. Belotti, J. Lee, L. Liberti, F. Margot, and A. Wächter. Branching and bounds tightening techniques for non-convex MINLP. *Optimization Methods and Software*, 24(4):597–634, 2009.

[24] A. Ben-Tal and A. Nemirovski. *Lectures on Modern Convex Optimization: analysis, algorithms, and engineering applications*. MPS-SIAM Series on Optimization. Society for Industrial and Applied Mathematics, Philadelphia, 2001.

[25] A. Ben-Tal, G. Eiger, and V. Gershovitz. Global minimization by reducing the duality gap. *Mathematical Programming*, 63(1):193–212, 1994.

[26] J.M. Bloemhof-Ruwaard and E.M.T. Hendrix. Generalized bilinear programming: An application in farm management. *European Journal of Operational Research*, 90(1): 102–114, 1996.

[27] C. Buchheim, A. Caprara, and A. Lodi. An effective branch-and-bound algorithm for convex quadratic integer programming. In *Integer Programming and Combinatorial Optimization: 14th International Conference, IPCO*, volume 6080, pages 285–298, Lausanne, Switzerland, 2010. Springer.

[28] M.R. Bussieck, A.S. Drud, and A. Meeraus. MINLPLib - A collection of test models for mixed-integer nonlinear programming. *INFORMS Journal on Computing*, 15(1): 114–119, 2003.

[29] A. Caprara and M. Monaci. Bidimensional packing by bilinear programming. *Mathematical Programming*, 118(1):75–108, 2009.

[30] S. Ceria and J. Soares. Convex programming for disjunctive convex optimization. *Mathematical Programming*, 86(3):595–614, 1999.

[31] S. Ceria, C. Cordier, H. Marchand, and L.A. Wolsey. Cutting planes for integer programs with general integer variables. *Mathematical programming*, 81(2):201–214, 1998.

[32] T. Christof and A. Löbel. PORTA: POlyhedron Representation Transformation Algorithm, 2009. URL `http://www.zib.de/Optimization/Software/Porta/`. May 2012.

[33] K. Chung, J.P.P. Richard, and M. Tawarmalani. Lifted inequalities for 0-1 mixed-integer bilinear covering sets, 2011. URL `http://www.optimization-online.org/DB\_HTML/2011/03/2949.html`. Optimization Online, May 2012.

[34] H. Crowder, E.L. Johnson, and M. Padberg. Solving large-scale zero-one linear programming problems. *Operations Research*, 31:803–834, 1983.

[35] C. D'Ambrosio, A. Frangioni, L. Liberti, and A. Lodi. Experiments with a feasibility pump approach for nonconvex MINLPs. In P. Festa, editor, *Proceedings of the 9th Symposium on Experimental Algorithms (SEA 2010)*, volume 6049 of *Lecture Notes in Computer Science*, pages 350–360. Springer, 2010.

[36] C. D'Ambrosio, J. Linderoth, and J. Luedtke. Valid inequalities for the pooling problem with binary variables. In Oktay Günlük and Gerhard Woeginger, editors, *Proceedings of the 15th international conference on Integer programming and combinatorial optimization*, Lecture Notes in Computer Science, pages 117–129, 2011.

[37] C.W. DeWitt, L.S. Lasdon, A.D. Waren, D.A. Brenner, and S.A. Melhem. Omega: An improved gasoline blending system for texaco. *Interfaces*, pages 85–101, 1989.

[38] J. Edmonds. Submodular functions, matroids, and certain polyhedra. In M. Jünger, G. Reinelt, and G. Rinaldi, editors, *Combinatorial optimization - Eureka, you shrink!*, chapter 2, pages 11–26. Springer-Verlag, 2003.

[39] J.E. Falk and R.M. Soland. An algorithm for separable nonconvex programming problems. *Management Science*, 15:550–569, 1969.

[40] C.A. Floudas and A. Aggarwal. A decomposition strategy for global optimum search in the pooling problem. *ORSA Journal on Computing*, 2(3):225–235, 1990.

[41] C.A. Floudas and I.E. Grossmann. Synthesis of flexible heat exchanger networks with uncertain flowrates and temperatures. *Computers & Chemical Engineering*, 11(4): 319–336, 1987.

[42] C.A. Floudas and V. Visweswaran. A global optimization algorithm (GOP) for certain classes of nonconvex NLPs–I. Theory. *Computers and Chemical Engineering*, 14(12): 1397–1417, 1990.

[43] L.R. Foulds, D. Haugland, and K. Jörnsten. A bilinear approach to the pooling problem. *Optimization*, 24(1):165–180, 1992.

[44] A.S. Freire, E. Moreno, and J.P. Vielma. An integer linear programming approach for bilinear integer programming. *Operations Research Letters*, 40(2):74–77, 2012.

[45] L. Frimannslund, G. Gundersen, and D. Haugland. Sensitivity analysis applied to the pooling problem. Technical Report 380, University of Bergen, December 2008.

[46] K.C. Furman and I.P. Androulakis. A novel MINLP-based representation of the original complex model for predicting gasoline emissions. *Computers & Chemical Engineering*, 32(12):2857–2876, 2008.

[47] GAMS. The General Algebraic Modeling System, 2011. URL `http://www.gams.com/docs/document.htm`.

[48] P.E. Gill, W. Murray, and M.A. Saunders. SNOPT: An SQP algorithm for large-scale constrained optimization. *SIAM Journal on Optimization*, 12(4):979–1006, 2002.

[49] F. Glover. Improved linear integer programming formulations of nonlinear integer problems. *Management Science*, 22(4):455–460, 1975.

[50] C.E. Gounaris, R. Misener, and C.A. Floudas. Computational comparison of piecewise-linear relaxations for pooling problems. *Industrial & Engineering Chemistry Research*, 48(12):5742–5766, 2009.

[51] H. Greenberg. Analyzing the pooling problem. *ORSA Journal on Computing*, 7(2): 205–217, 1995.

[52] Z. Gu, G.L. Nemhauser, and M.W.P. Savelsbergh. Lifted cover inequalities for 0-1 integer programs: Computation. *INFORMS Journal on Computing*, 10:427–437, 1998.

[53] Z. Gu, G.L. Nemhauser, and M.W.P. Savelsbergh. Sequence independent lifting in mixed integer programming. *Journal of Combinatorial Optimization*, 4(1):109–129, 2000.

[54] O. Günlük, J. Lee, and J. Leung. A polytope for a product of real linear functions in 0/1 variables. In J. Lee and S. Leyffer, editors, *Mixed Integer Nonlinear Programming*, volume 154 of *IMA Volumes in Mathematics and its Applications*, pages 513–531. Springer, 2012.

[55] P. Hansen and B. Jaumard. Reduction of indefinite quadratic programs to bilinear programs. *Journal of Global optimization*, 2(1):41–60, 1992.

[56] I. Harjunkoski, T. Westerlund, R. Pörn, and H. Skrifvars. Different transformations for solving non-convex trim loss problems by MINLP. *European Journal of Operational Research*, 105:594–603, 1998.

[57] M.M. Hasan and I.A. Karimi. Piecewise linear relaxation of bilinear programs using bivariate partitioning. *AIChE Journal*, 56(7):1880–1893, 2010.

[58] MM Hasan and IA Karimi. Piecewise linear relaxation of bilinear programs using bivariate partitioning. *AIChE Journal*, 56(7):1880–1893, 2010.

[59] D. Haugland. An overview of models and solution methods for pooling problems. *Energy, Natural Resources and Environmental Economics*, pages 459–469, 2010.

[60] C.A. Haverly. Studies of the behavior of recursion for the pooling problem. *ACM SIGMAP Bulletin*, 25:19–28, 1978.

[61] R. Horst and H. Tuy. *Global optimization : Deterministic approaches*. Springer, 3rd edition, 2003.

[62] I. ILOG. IBM ILOG CPLEX 12.2 User's Manual. IBM ILOG, Gentilly, France, 2009.

[63] J. Kallrath. Mixed integer optimization in the chemical process industry: Experience, potential, and future perspectives. *Trans IChemE*, 78(6):809–822, 2000.

[64] R. Karuppiah and I.E. Grossmann. Global optimization for the synthesis of integrated water systems in chemical processes. *Computers and Chemical Engineering*, 30(4): 650–673, 2006.

[65] M. Laurent and A. Sassano. A characterization of knapsacks with the max-flow-min-cut property. *Operations Research Letters*, 11:105–110, 1992.

[66] S. Lee and I.E. Grossmann. Global optimization of nonlinear generalized disjunctive programming with bilinear equality constraints: applications to process networks. *Computers & chemical engineering*, 27(11):1557–1575, 2003.

[67] L. Liberti and C.C. Pantelides. An exact reformulation algorithm for large nonconvex NLPs involving bilinear terms. *Journal of Global Optimization*, 36(2):161–189, 2006.

[68] Q. Louveaux and L.A. Wolsey. Lifting, superadditivity, mixed integer rounding and single node flow sets revisited. *4OR: A Quarterly Journal of Operations Research*, 1 (3):173–207, 2003.

[69] J. Luedtke, M. Namazifar, and J. Linderoth. Some results on the strength of relaxations of multilinear functions. Technical report, University of Wisconsin-Madison, 2010. URL `www.optimization-online.org/DB\_FILE/2010/08/2711.pdf`. Optimization Online, June 2012.

[70] G.P. McCormick. Computability of global solutions to factorable nonconvex programs: Part I. convex underestimating problems. *Mathematical Programming*, 10(1):147–175, 1976.

[71] C.A. Meyer and C.A. Floudas. Convex envelopes for edge-concave functions. *Mathematical programming*, 103(2):207–224, 2005.

[72] C.A. Meyer and C.A. Floudas. Global optimization of a combinatorially complex generalized pooling problem. *AIChE journal*, 52(3):1027–1037, 2006.

[73] R. Misener and C.A. Floudas. MINLP library : Generalized pooling problem, 2011. URL `http://www.minlp.org/problems/ver/159/results/Results\_Discussion\_GeneralizedPooling.pdf`. July 2012.

[74] R. Misener and C.A. Floudas. Advances for the pooling problem: Modeling, global optimization, and computational studies. *Appl. Comput. Math*, 8(1):3–22, 2009.

[75] R. Misener and C.A. Floudas. Global optimization of large-scale generalized pooling problems: Quadratically constrained MINLP models. *Industrial & Engineering Chemistry Research*, 49(11):5424–5438, 2010.

[76] R. Misener and C.A. Floudas. Global optimization of mixed-integer quadratically-constrained quadratic programs (MIQCQP) through piecewise-linear and edge-concave relaxations. *Math. Program. B,. Accepted for Publication*, 2011.

[77] R. Misener, C.E. Gounaris, and C.A. Floudas. Mathematical modeling and global optimization of large-scale extended pooling problems with the (EPA) complex emissions constraints. *Computers & Chemical Engineering*, 34(9):1432–1456, 2010.

[78] R. Misener, J.P. Thompson, and C.A. Floudas. APOGEE: Global optimization of standard, generalized, and extended pooling problems via linear and logarithmic partitioning schemes. *Computers & Chemical Engineering*, 35:876–892, 2011.

[79] G.L. Nemhauser and L.A. Wolsey. *Integer and combinatorial optimization*, volume 18. Wiley New York, 1988.

[80] T. Nishi. A semidefinite programming relaxation approach for the pooling problem. Master's thesis, Department of Applied Mathematics and Physics, Kyoto University, February 2010. URL `http://www-optima.amp.i.kyoto-u.ac.jp/result/masterdoc/21nishi.pdf`.

[81] J.H. Owen and S. Mehrotra. On the value of binary expansions for general mixed-integer linear programs. *Operations Research*, 50(5):810–819, 2002.

[82] V. Pham, C. Laird, and M. El-Halwagi. Convex hull discretization approach to the global optimization of pooling problems. *Industrial and Engineering Chemistry Research*, 48(4):1973–1979, 2009.

[83] Y. Pochet and R. Weismantel. The sequential knapsack polytope. *SIAM Journal on Optimization*, 8(1):248–264, 1998.

[84] M.B. Poku, L.T. Biegler, J.D. Kelly, R. Coxhead, and V. Gopal. Nonlinear programming algorithms for large nonlinear gasoline blending problems. In I.E. Grossmann and C. McDonald, editors, *Foundations of Computer-Aided Process Operations*, Coral Springs, Florida, 2003.

[85] I. Quesada and I.E. Grossmann. Global optimization of bilinear process networks with multicomponent flows. *Computers and Chemical Engineering*, 19(12):1219–1242, 1995.

[86] M. Realff, S. Ahmed, H. Inacio, and K. Norwood. Heuristics and upper bounds for a pooling problem with cubic constraints. In *Foundations of Computer-Aided Process Operations*, Savannah, GA, January 2012.

[87] J.P.P. Richard and M. Tawarmalani. Lifting inequalities: a framework for generating strong cuts for nonlinear programs. *Mathematical Programming*, 121(1):61–104, 2010.

[88] J.P.P. Richard, IR de Farias Jr, and GL Nemhauser. Lifted inequalities for 0-1 mixed integer programming: Basic theory and algorithms. *Mathematical Programming*, 98 (1):89–113, 2003.

[89] A.D. Rikun. A convex envelope formula for multilinear functions. *Journal of Global Optimization*, 10(4):425–437, 1997.

[90] G. Rinaldi, U. Voigt, and G.J. Woeginger. The mathematics of playing golf, or: a new class of difficult non-linear mixed integer programs. *Mathematical Programming*, 93(1):77–86, 2002.

[91] J.P. Ruiz and I.E. Grossmann. Exploiting vector space properties to strengthen the relaxation of bilinear programs arising in the global optimization of process networks. *Optimization Letters*, 5(1):1–11, 2011.

[92] M. Ruiz, O. Briant, J.M. Clochard, and B. Penz. Large-scale standard pooling problems with constrained pools and fixed demands. *Journal of Global Optimization*, pages 1–18, 2012.

[93] N.V. Sahinidis. BARON: A general purpose global optimization software package. *Journal of Global Optimization*, 8(2):201–205, 1996.

[94] A. Saxena, P. Bonami, and J. Lee. Convex relaxations of non-convex mixed integer quadratically constrained programs: projected formulations. *Mathematical programming*, 130(2):359–413, 2011.

[95] P.D. Seymour. The matroids with the max-flow min-cut property. *Journal of Combinatorial Theory, Series B*, 23(2-3):189–222, 1977.

[96] H.D. Sherali. Convex envelopes of multilinear functions over a unit hypercube and over special discrete sets. *Acta mathematica vietnamica*, 22(1):245–270, 1997.

[97] H.D. Sherali and W.P. Adams. *A reformulation-linearization technique for solving discrete and continuous nonconvex problems*, volume 31 of *Nonconvex Optimization and its Applications*. Kluwer Academic Publishers, 1998.

[98] H.D. Sherali and A. Alameddine. A new reformulation-linearization technique for bilinear programming problems. *Journal of Global Optimization*, 2(4):379–410, 1992.

[99] H.D. Sherali, W.P. Adams, and P.J. Driscoll. Exploiting special structures in constructing a hierarchy of relaxations for 0-1 mixed integer problems. *Operations Research*, 46(3):396–405, 1998.

[100] F. Tardella. Existence and sum decomposition of vertex polyhedral convex envelopes. *Optimization Letters*, 2(3):363–375, 2008.

[101] M. Tawarmalani and N.V. Sahinidis. *Convexification and global optimization in continuous and mixed-integer nonlinear programming: theory, algorithms, software, and applications*, chapter 9, pages 281–310. Kluwer Academic Publishers, 2002.

[102] M. Tawarmalani and N.V. Sahinidis. Convex extensions and envelopes of lower semi-continuous functions. *Mathematical Programming*, 93(2):247–263, 2002.

[103] M. Tawarmalani and N.V. Sahinidis. *Convexification and global optimization in continuous and mixed-integer nonlinear programming: theory, algorithms, software, and applications*. Kluwer Academic Publishers, 2002.

[104] M. Tawarmalani and N.V. Sahinidis. A polyhedral branch-and-cut approach to global optimization. *Mathematical Programming*, 103(2):225–249, 2005.

[105] M. Tawarmalani, J-P.P. Richard, and K. Chung. Strong valid inequalities for orthogonal disjunctions and bilinear covering sets. *Mathematical Programming*, 124:481–512, 2010.

[106] D. Vandenbussche and G.L. Nemhauser. A branch-and-cut algorithm for nonconvex quadratic programs with box constraints. *Mathematical Programming*, 102(3):559–575, 2005.

[107] L.N. Vicente, P.H. Calamai, and J.J. Júdice. Generation of disjointly constrained bilinear programming test problems. *Computational Optimization and Applications*, 1(3):299–306, 1992.

[108] J.P. Vielma and G.L. Nemhauser. Modeling disjunctive constraints with a logarithmic number of binary variables and constraints. *Mathematical Programming*, 128:49–72, 2011.

[109] J.P. Vielma, S. Ahmed, and G.L. Nemhauser. A lifted linear programming branch-and-bound algorithm for mixed-integer conic quadratic programs. *INFORMS Journal on Computing*, 20(3):438–450, 2008.

[110] V. Visweswaran. MINLP: Applications in blending and pooling problems. In C.A. Floudas and P.M. Pardalos, editors, *Encyclopedia of Optimization*, pages 2114–2121. Springer, 2009.

[111] V. Visweswaran and C.A. Floudas. A global optimization algorithm (GOP) for certain classes of nonconvex NLPs–II. Application of theory and test problems. *Computers & chemical engineering*, 14(12):1419–1434, 1990.

[112] A. Wächter and L.T. Biegler. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical Programming*, 106(1):25–57, 2006.

[113] D.S. Wicaksono and IA Karimi. Piecewise MILP under-and overestimators for global optimization of bilinear programs. *AIChE Journal*, 54(4):991–1008, 2008.

[114] L.A. Wolsey. Facets and strong valid inequalities for integer programs. *Operations Research*, 24:367–372, 1976.

[115] LA Wolsey. Valid inequalities and superadditivity for 0-1 integer programs. *Mathematics of Operations Research*, 2:66–77, 1977.

[116] L.A. Wolsey. Strong formulations for mixed integer programs: valid inequalities and extended formulations. *Mathematical programming*, 97(1):423–447, 2003.

[117] J.M. Zamora and I.E. Grossmann. A branch and contract algorithm for problems with concave univariate, bilinear and linear fractional terms. *Journal of Global Optimization*, 14(3):217–249, 1999.